Ferry check-in with NFC or 2D barcode

Sebastian Hew

Degree project in
Computer Science
Second cycle
Stockholm, Sweden 2013
DEGREE PROJECT
AT CSC, KTH

Ferry check-in with NFC or 2D barcode

Incheckning på färja med NFC eller 2D-streckkod

Hew, Sebastian
E-postadress vid KTH: hew@kth.se
Exjobb i: Datalogi
Handledare: Arvestad, Lars
Examinator: Lagergren, Jens
Uppdragsgivare: Softronic
2013-06-25
Ferry check-in with NFC or 2D barcode

Abstract

The increased use of smartphones in recent years has opened up the opportunity to modernize manual operations, such as checking-in at ferry terminals and airports. Many airlines have been using 2D barcodes for a while and are planning to support NFC in the coming years. Destination Gotland wants their passengers to be able to check-in at their ferry terminals with both of these technologies.

In this thesis, I propose a system where passengers are able to check-in with their smartphones, present a prototype and analyze any advantages. Wait times are estimated with queueing theory and the results show that the wait times for passengers will decrease with the proposed system.

Incheckning på färja med NFC eller 2D-streckkod

Sammanfattning

Det ökande användandet av smartphones under senare år har öppnat upp möjligheten att modernisera manuella operationer, så som incheckning i färjeterminaler eller på flygplatser. Många flygbolag har redan använt 2D-streckkoder ett tag och planerar nu att införa stöd för NFC inom de kommande åren. Destination Gotland vill att deras passagerare ska kunna checka in på deras färjeterminaler med båda teknologierna.

I detta examensarbete föreslår jag ett system där passagerare kan checka in med sina smartphones, jag presenterar en prototyp och jag analyserar dess fördelar. Kötider uppskattas med hjälp utav köteori och resultaten visar att kötiderna för passagerarna minskar med det föreslagna systemet.
Acknowledgement

I would like to thank all the people at Softronic who helped making this project possible, mainly Mattias Lindström who helped me define the specification during the early stage and my supervisor, Benjamin Özmen, for all the feedback and ideas during our weekly meetings.

I would also like to thank Destination Gotland for allowing me the opportunity of doing this project and the staff at the Nynäshamn terminal for being extremely welcoming and helpful during my two visits there.

Last but not least, I want to thank my supervisor at CSC, Lars Arvestad, and my examiner Jens Lagergren.
# Table of Contents

1  Introduction ........................................................................................................................................ 1  
1.1  Project background ....................................................................................................................... 1  
1.2  Problem statement and scope .......................................................................................................... 1  
1.3  Method ............................................................................................................................................. 2  
1.4  Abbreviations ................................................................................................................................. 2  

2  Theory .................................................................................................................................................. 4  

2.1  Background ..................................................................................................................................... 4  
2.1.1  Mobile phones ............................................................................................................................. 4  
2.1.2  Barcodes ..................................................................................................................................... 5  
2.1.3  RFID ......................................................................................................................................... 6  
2.1.4  Smart cards ............................................................................................................................... 7  
2.2  NFC .................................................................................................................................................. 8  
2.2.1  Operating modes ....................................................................................................................... 8  

2.3  Previous work .................................................................................................................................. 10  
2.3.1  Air travel ................................................................................................................................. 10  
2.3.2  Public transportation ................................................................................................................ 10  
2.3.3  Real-world implementations and trials .................................................................................... 11  

2.4  Queuing theory ............................................................................................................................... 12  
2.4.1  Poisson process ....................................................................................................................... 12  
2.4.2  Formulas .................................................................................................................................. 13  

3  Implementation .................................................................................................................................. 15  

3.1  Hardware ....................................................................................................................................... 15  
3.1.1  Why Android ............................................................................................................................ 15  
3.2  Android development ..................................................................................................................... 15  
3.2.1  Activity ..................................................................................................................................... 15  
3.2.2  Intent ....................................................................................................................................... 16  
3.2.3  NFC ....................................................................................................................................... 16  
3.2.4  Barcodes ............................................................................................................................... 17  

3.3  Application ..................................................................................................................................... 17  
3.3.1  First attempt ............................................................................................................................ 17
3.3.2 Final prototype ................................................................. 18
4 Results .................................................................................. 21
  4.1 Check-in process ................................................................. 21
    4.1.1 Discounts .................................................................. 21
    4.1.2 Pre-check-in .............................................................. 22
    4.1.3 Time consuming activities during check-in ................. 22
  4.2 Survey ................................................................................. 23
    4.2.1 Calculations on current processes ............................... 24
  4.3 Application ........................................................................ 25
  4.4 Calculations for proposed system .................................... 30
5 Discussion .............................................................................. 32
  5.1 Service time ..................................................................... 32
  5.2 Other improvements ........................................................ 33
  5.3 Sources of error ............................................................... 34
  5.4 Conclusions ...................................................................... 35
References .................................................................................. 36
Appendices .................................................................................. 39
  A Survey ................................................................................ 39
1 Introduction

1.1 Project background

This project was performed at Softronic on behalf of their client Destination Gotland. Destination Gotland operates all ferry traffic between the island of Gotland and the Swedish mainland on behalf of the Swedish government. Every year, approximately 1.6 million passengers travel to or from Gotland with one of their ferries. Apart from the harbor in Visby, Gotland there are also harbors in Nynäshamn and Oskarshamn on the mainland. They currently use four different ships, with capacities ranging from 700-1500 passengers and 145-500 cars. Softronic is responsible for Destination Gotland’s web portal and online booking.

The check-in opens 1.5-2 hours before departure and it is recommended to be there in good time in order to avoid queues, which indicates that there is room for improvement. The check-in is a manual operation, the customer show their travel documents or provides a booking number and will then be given a ticket with a seat number.

Barcode scanning and Near Field Communication (NFC) are technologies which have grown more and more common with the increased usage of smartphones. NFC allows users to wirelessly exchange data by tapping two phones together. It only works in close range and the pairing between devices is automatic, as opposed to other wireless technologies such as Bluetooth. NFC and barcode scanning open up the possibility for more convenient check-in options.

1.2 Problem statement and scope

The aim of this thesis was to analyze the current check-in process at Destination Gotland and propose how it can be improved with the use of said mobile technologies. Is it possible to speed up the process with the use of barcodes and NFC during the check-in? Are there other features which could be implemented that could speed up the process? Are there other advantages with the introduction of barcodes and NFC during the check-in? The analysis was performed by observing current routines during check-in. A prototype application which can perform basic check-in operations was developed. This thesis will focus mainly on the use of NFC, although QR codes are also included since not all phones have support for NFC.

In order to have a clear view of the scope for this project it is important to define delimitations. The delimitations for this project are:

- The prototype application will only be developed for the Android platform.
- The prototype application will only use QR code and NFC for the check-in operation.
The application will not be released and evaluated at real-world check-ins during this project, but rather by estimations and assumptions compared with the observable data from the study.

1.3 Method
The study included several trips to the Nynäshamn terminal. The first step of the study was to examine current processes and routines during check-in and find out if there are parts of it that are more time-consuming than others. The next step, after having developed the prototype application was to determine what parts of the process that could be improved and by approximately how much. The analysis was performed with the help of queueing theory, which means that a deterministic estimation of the queueing time was calculated. Service times and arrival rates are needed for the calculation, and they were estimated at the terminal.

Things that were noted at the terminal:

- Detailed description of the check-in process
- Queueing theory
  - Service times
  - Arrival rates
- Special cases where the service time is longer than usual

The main goal of the first trip to Nynäshamn was to compose a detailed description of the check-in process. It was done partially by observing the check-in, but also by talking to the Destination Gotland employees at the terminal.

1.4 Abbreviations
A list of the abbreviations that will be used throughout this report.

API Application Programming Interface
GSM Global System for Mobile Communications
GSMA GSM Association
IATA International Air Transport Association
IC Integrated Circuit
LLCP Logical Link Control Protocol
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMS</td>
<td>Multimedia Messaging Service</td>
</tr>
<tr>
<td>NDEF</td>
<td>NFC Data Exchange Format</td>
</tr>
<tr>
<td>NFC</td>
<td>Near Field Communication</td>
</tr>
<tr>
<td>NPP</td>
<td>NDEF Push Protocol</td>
</tr>
<tr>
<td>OTA</td>
<td>Over-the-air</td>
</tr>
<tr>
<td>QR</td>
<td>Quick Response</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identification</td>
</tr>
<tr>
<td>SE</td>
<td>Secure Element</td>
</tr>
<tr>
<td>SMC</td>
<td>Secure Memory Card</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Messaging Service</td>
</tr>
<tr>
<td>SNEP</td>
<td>Simple NDEF Exchange Protocol</td>
</tr>
<tr>
<td>UICC</td>
<td>Universal Integrated Circuit Card</td>
</tr>
<tr>
<td>UPC</td>
<td>Universal Product Code</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>

*Table 1* List of abbreviations
2 Theory

The two technologies that will be used for this project are NFC and barcode scanning. NFC, or Near Field Communication, is a technology which allows two devices to exchange data by touching them together. Barcode scanning is an older technology, perhaps most famously known for its use by stores for keeping track of inventory. This chapter will provide a theoretical background to this project, with a focus on NFC. It will also present previous work similar to this project, such as the use of NFC and barcodes at airports. And lastly, it will give a short introduction to queueing theory, which will be used later to analyze the results from the conducted study.

2.1 Background

NFC has its roots in several related technologies and is considered to be a synergy of technologies such as mobile devices, RFID communication and smart cards (Coskun, et al., 2012, p. 41). Figure 1 shows the technologies which have directly or indirectly influenced NFC. The following sections will give a short explanation of each of these parts.

![Evolution of NFC technology](image)

**Figure 1** Evolution of NFC technology (Adapted from Coskun, et al., 2012, p. 43)

2.1.1 Mobile phones

NFC is closely tied to ubiquitous computing, a model where machines should be designed to fit human needs instead of forcing humans to adjust their activities to fit the machines. The aim is to allow humans to perform activities with machines without changing their normal behavior and not necessarily be aware that they are using machines. Mobile phones play a big part in achieving ubiquitous computing, with the rise of smartphones and the use of touch-screen input to lead the way.
The initial purpose of mobile phones was to enable voice communication between users. With GSM, mobile phones were able to communicate with voice as well as SMS, MMS and limited Internet access. In the early 2000s, Bluetooth technology became very popular, which enabled mobile phones to connect wirelessly to accessories such as headsets and exchange data between other Bluetooth devices. Bluetooth was not considered a safe way to exchange data however, since you could connect to devices within a range of tens of meters which allowed malicious devices to tap the communication (Coskun, et al., 2012, p. 42). The long range of Bluetooth also has another disadvantage; there may be several Bluetooth devices within range so the user has to manually select which device to connect to from a list. NFC on the other hand, does not have these problems. The close range makes it really difficult to tap the communication, and the pairing between devices is also much smoother since the user simply has to touch the device to a tag or another device to start the communication.

2.1.2 Barcodes

Traditional barcodes were first used commercially 1974 when a UPC code on a pack of Wrigley’s gum was scanned in Ohio (Collins, 2009). Universal Product Code (UPC) is a one-dimensional barcode, which means that they are scanned only in one direction. The downside of one-dimensional barcodes are that their storing capabilities are very limited, generally no more than 20 characters can be stored in them, and thus there was a need for a new type of barcode (Winter, 2011, pp. 17-18).

QR codes were invented in 1994 by the Toyota subsidiary Denso Wave for the purpose of keeping track of inventory. During the development of the QR code, Denso first experimented with stacking a few one-dimensional barcodes on top of each other. Their next version was a combination of one-dimensional barcodes and two-dimensional barcodes. The final version is the QR-code as we know it today (Winter, 2011, p. 117). This evolution is depicted in Figure 2.

![Figure 2](image_url)  
**Figure 2** Evolution of the QR code (Adapted from Winter, 2011, p. 117)

QR codes are the most commonly used two-dimensional barcode for personal use. There are many uses for QR codes, for example product information (price, country of origin, manufacturing information), navigation and schedules for public transportation, business cards, tour information in museums and zoos, and much more.

A QR code uses specific patterns which are the same for all QR codes. Figure 3 shows the structure of a QR code, including these required patterns. The three squares at the corners are used for determining the size of the code and positional
data such as direction and angle. The alignment pattern makes the code easier to scan when it is placed on round surfaces. The timing pattern helps with detecting the data cells (Sinha, 2012).

Figure 3 Structure of a QR code (With permission Wheeler, 2008)

The storage capacity of QR code will vary depending on its size. Adding more rows and columns to the code will increase the amount of data it can store. The largest size is known as Version 40, which has 177 rows and 177 columns and is able to store up to 4296 alphanumerical characters (Denso Wave, n.d.).

2.1.3 RFID

RFID stands for Radio-Frequency Identification and is a technology to wirelessly exchange data between an RFID reader and an RFID tag by using radio waves. NFC uses the same radio technology. The RFID reader can read/write data from/to RFID tags at varying ranges. The technology was first used during the Second World War, to identify an aircraft as friend or foe. Commercial usage began in the 1960s and 1970s with door key opening systems (Coskun, et al., 2012, p. 51).

RFID tags contain an IC and an antenna. The IC holds the data that will be transmitted to the reader. The antenna receives and transmits the RFID signal. Tags can either be passive or active. The main difference between an active and a passive tag is that the passive tag does not have its own power supply; instead the power will be supplied by the reader. Passive tags have a shorter range than active tags, varying between 10 cm to a few meters, which changes depending on radio frequency, antenna design and size. Active tags have their own power supply. There are a few advantages that come with having an embedded power source, like having a much longer range and larger memory size. They are also more effective in RF challenged environments such as underwater and in shipping containers (Coskun, et al., 2012, p. 55). The drawbacks being that they are more expensive and physically larger than passive tags.

RFID readers contain a transceiver, a control unit and an antenna. The antenna emits radio waves to initiate the data exchange, the reader then captures the transmitted data which is interpreted by a decoder in the transceiver.

The two most important operating principles of RFID are inductive coupling and backscatter coupling (Finkenzeller, 2010). Inductive coupling is often used on
passive tags. The reader’s antenna generates a high frequency electromagnetic field. The tag will draw energy when it is placed inside the electromagnetic field, which leads to a voltage drop at the internal resistance in the reader’s antenna. The tag can switch on and off a load resistance which will lead to voltage changes at the reader’s antenna. When using inductive coupling, the data is thus transferred by controlling when the load resistance is switched on and off.

When using backscatter coupling the reader’s antenna sends out an electromagnetic field. When the field hits the tag’s antenna, it will reflect back some of the power from the reader. To send data back to the reader, the tag uses a load resistor which is switched on and off to alter the data that is reflected back to the reader.

2.1.4 Smart cards

Smart cards date back to the invention of magnetic stripe cards by IBM in the 1960s. They use a stripe of magnetic particles attached to a card and the data that is stored on them can be read by swiping them on a device with a magnetic reader. The first smart cards were used for telephone payments in the 1980s. In the 1990s, GSM based mobile phones started using smart card SIMs.

A smart card is a card with an embedded integrated circuit (IC), which stores the information used by the smart card. There are three kinds of smart cards: ones that connects physically to a reader (contact smart card), ones that connect with an RF interface (contactless smart card) as well as hybrid smart cards which make use of both technologies.

Contact smart cards have a, usually gold plated, IC card placed on the surface of the smart card. The IC card makes physical contact with a reader in order to transmit data. These IC cards are often found on bank cards and SIM-cards for mobile phones. Figure 4 shows a contact smart card with the gold plated IC card on the left side.

![A Contact Smart Card](image)

**Figure 4** A Contact Smart Card

Contactless smart cards have an IC chip, which stores the data, and an antenna. The antenna is used to draw power from the reader as well as exchange data. Some of the most common usages of contactless smart cards are public transportation, for example the SL Access system used in Stockholm, and employee access cards used on workplaces.
2.2 NFC

NFC technology was developed jointly by Philips and Sony in 2002. The communication is bidirectional between two devices within a few centimeters. The technology and its specifications are primarily developed by the NFC Forum, which is a non-profit industry association which was founded in 2004 by Nokia, Philips and Sony. The forum now has over 170 members, including manufacturers, application developers and financial service institutions. The goal of the NFC Forum is to develop specifications to encourage the development and educate the consumers about NFC. There are three different NFC devices: NFC enabled mobile phones, NFC tags and NFC readers and they may operate in one of three different modes, readerwriter, peer-to-peer and card emulation.

NFC uses the same technology as RFID and is regarded as an extension to it. The key differences between the two are that NFC only works at short range and that it allows for two-way communication. NFC uses a frequency of 13.56 MHz and a data rate of no more than 424 kbps.

Just like RFID, NFC devices can be in either active or passive mode. Active devices have their own embedded power source while the passive devices get their power from the RF field from the other device. The device that starts the communication is called the initiator and the device that responds is called the target. An active device may act as both an initiator and a target, while a passive device can only be a target. NFC tags are just a passive RFID tag. NFC tags cannot communicate with other NFC tags, since they need a power source, an initiator, in order to start the communication.

2.2.1 Operating modes

Readerwriter mode

In this mode, an active NFC device reads or writes data from or to an NFC tag. The active device could be either an NFC reader or an NFC enabled mobile phone; however the mobile phone is more common in this case. Since the NFC tag is passive, the active device acts as the initiator.

A mobile phone will be able to perform various actions after reading a tag, for example start a web browser and load the page with the URL that is stored on the tag. The most common usage of the readerwriter mode is smart posters, which are advertising posters with NFC tags containing for example URLs or coupons (Coskun, et al., 2012, p. 120).

Readerwriter mode uses the NFC Forum standard NDEF to exchange data between devices. An NDEF message consists of one or more NDEF records, which can be chained together to create larger payloads. A record consists of a payload and three parameters to describe the payload: payload length, payload type and a payload identifier as described in Figure 5.
**Peer-to-peer mode**

When using peer-to-peer, two active NFC devices are able to exchange data between each other. The two devices could be two NFC enabled mobile phones, but NFC readers can also be used to for example exchange data between a phone and a computer connected to the reader. Since both devices are active any of the two may act as the initiator. An example of usage for peer-to-peer mode is exchanging business cards or digital photos.

The NFC Forum has defined two protocols for exchanging data over peer-to-peer mode; Simple NDEF Exchange Protocol (SNEP) and Logical Link Protocol (LLCP). SNEP allows two devices to exchange NDEF messages when operating in peer-to-peer mode. SNEP makes use of LLCP for the exchange. LLCP is a lower level protocol (layer 2 on the OSI model, which is the data link layer).

**Card emulation mode**

This mode allows an active NFC device to act as an NFC tag (in other words, a contactless smart card, hence the name). The device may then interact with another active device in reader/writer mode. Since the device emulates an NFC tag, it acts as a passive device which means that the reader will be the initiator and provide the RF field. The biggest usage for card emulation is payment, to make phones act as a debit or credit card. Other usages may be ticketing, for public transportation, theatres, flight tickets etc; and access control, for hotel rooms, cars etc.

Card emulation is the only one of the three modes that uses a secure element (SE). The SE is a combination of hardware, software, interfaces and protocols. Using an SE correctly assures the users that the process is being performed in a secure manner. There are three popular options for an SE (Coskun, et al., 2012, pp. 13-14):

- Embedded Hardware, a non-removable part of the mobile phone, which means that the chip is being installed during the manufacturing of the phone.

---

**Figure 5** Structure of an NDEF message (Adapted from Nokia, 2012)
• Secure Memory Card (SMC), a removable memory card, for example an SD-card with an embedded SE.

• Universal Integrated Circuit Card (UICC), a card on which the SIM is implemented on. The UICC is also a removable part of the mobile phone. Coskun, et al. (2012, p. 15) labels the UICC as an ideal environment for NFC applications since it is personal, secure and portable. It is also easy to remotely install applications on the UICC with over-the-air (OTA) technology.

2.3 Previous work

There are many whitepapers surrounding NFC for travel and transportation, however none that are specifically targeted towards ferry boarding. I decided to take a closer look on one whitepaper targeted towards air travel (GSMA, IATA, 2011) and another one towards public transportation in general (NFC Forum, 2011). I have also looked at some real-world implementations and trials.

2.3.1 Air travel

The whitepaper about the benefits of NFC for air travel (GSMA, IATA, 2011) is collaboration between GSMA and IATA. GSMA is a trade association for mobile operators with nearly 800 members. IATA is a trade association for airlines with 240 members. According to a survey by IATA from 2009, 50% of all passengers would like more self-service options, indicating a demand for faster airport-common processes, such as check-in and boarding.

The whitepaper presents a couple of use cases a traveler might experience at an airport, namely: passenger check-in, baggage check-in, security checkpoint, lounge access, boarding and post-flight. Of course, only a few of these use cases are relevant for this thesis project, however the general consensus seems to be that NFC is easier to use and more convenient than a paper ticket. Barcodes, on a paper ticket or on a mobile phone, are more difficult to scan than simply tapping the phone on a reader as is the case with NFC.

2.3.2 Public transportation

The whitepaper about NFC in public transportation (NFC Forum, 2011) is produced by members of the NFC Forum, an association described in section 2.2. The whitepaper brings up benefits of NFC for travelers and transport operators as well as comparisons with other technologies.

The benefits for travelers, according to the whitepaper, are that storing tickets on the phone is more convenient than using paper tickets. Phones are more durable, less likely to be lost and more environmentally friendly. Phones also has advantages over plastic cards since an average person often has a wallet with several cards, and looking through your wallet for the right one is more time consuming than just using your phone.
The benefits for transport operators are increased traveler satisfaction, increased throughput and improved bus boarding times, which have been proved by existing programs. There are also reduced operating and maintenance costs by reducing the use of paper tickets.

Compared to paper magnetic stripe tickets, NFC is more reliable and convenient. Tickets are less likely to be lost. Obtaining tickets are also more convenient, since you do not have to visit a ticket office or ticket machine to get one. It is easier to manage several tickets on the phone compared to when using a wallet with contactless cards. An NFC-enabled phone is also less prone to collisions, as can be the case when trying to read a card from a wallet which holds several contactless cards. An NFC-enabled phone is also easier to read than a two-dimensional barcode, where the phones display has to be held up to the reader at a specific angle. Reading a barcode is a one-way communication; the phone can not receive any data from the reader. NFC is two-way, which means that the reader can tell the phone to open up an application or send back a receipt.

### 2.3.3 Real-world implementations and trials

QR codes and other 2D barcodes are already widely used for ticketing purposes. Jetstar is an airline which provides this feature (Jetstar, n.d.). Passengers will receive a QR boarding pass by mail if they register for automatic check-in or checks in online 48 hours prior to their flight. If they travel in a group, they may choose to receive one QR code for the whole party or a separate code for each passenger. If the passengers have luggage to check in, they can go to a self-service kiosk to scan their QR code, they will then receive bag tags and receive a boarding pass. If they do not have any luggage to check in, they can go directly to the gate to have their code scanned and receive a boarding pass.

Scandinavian Airlines (SAS) is another airline which allows passengers to check-in with a barcode (SAS, 2013). Passengers who want to use this feature will receive a link to their mobile boarding pass after checking-in via SMS or SAS mobile website. The mobile boarding pass will contain a 2D barcode as well as flight number, seating, departure time and arrival time.

Trials for NFC boarding have been conducted, but there are few implementations to date. Japan Airlines launched the first implementation for NFC boarding in October 2012 (Balaban, 2012). In 2009, a trial was conducted at the Nice Côte d’Azur Airport in France (Amadeus, 2009). The trial, which went under the name of Pass and Fly, was a joint project started by Air France and Nice Côte d’Azur Airport together with Amadeus and IER. Amadeus built the software while IER provided the hardware. The passengers identified themselves at an NFC reader, which retrieved their flight information from a database and uploaded a boarding pass to the passenger’s phone. The boarding pass could then be used at the security checkpoint and the boarding gate.

In Germany, there is a service called Touch&Travel (Touch&Travel, n.d.), which is an app smartphone users can download and use to pay when using public transportation. Examples of companies that provide this service for their customers are BVG and S-Bahn in Berlin, and RMV in the state of Hesse. After the users
have registered and installed the app, they can start their travel by registering at which station they start their trip. This can be done by scanning an NFC tag, reading a QR code, entering a code or even by registering their position. A ticket is then created in their app, which is shown during ticket inspections. When they have reached their destination they register at which station their trip ended, by again scanning an NFC tag or any of the other options.

2.4 Queuing theory

This section aims to give a brief introduction to queueing theory since the study will use it to analyze the results. Queueing theory gives a mathematical representation of queues. Queueing theory provides models from which formulas for predicting queue lengths and waiting times can be derived.

Customers arrive to get some sort of service. The service can for example be to pay at a grocery store, call customer support for your Internet service provider or, as in the case for this report, check-in at a ferry terminal. There can be one or several service stations, which means that more than one customer can possibly be served at the same time. If there is no service station available when a new customer arrives, then that customer has to wait in line. There are different models for the arrivals and the service times, for example they can be deterministic (arrivals are not random) or they can be random. It is probably most common to model the arrivals according to the Poisson process and use exponential distribution for the service time, which will be used in this report. This will provide arrivals and service times which are random.

![Figure 6 A queueing theory scenario](image)

Figure 6 shows a scenario where queueing theory could be applied. The squares represent the service stations and the circles represent the customers. The lower-most service station has just finished serving a customer and the first customer in the queue is on his way to that service-station, as the arrow in the image implies.

2.4.1 Poisson process

The Poisson process is a random process and is used to model random occurrences in time. Some common applications of the Poisson process is to model events such
as particle emission of a radioactive material, goals scored in a football match and, of course, the arrival of customers at a service station.

The time between an occurrence and the next one is defined as $\text{Exp}(\lambda)$, which means that it is exponentially distributed with a rate $\lambda$. The exponential distribution has an expected value of $\lambda^{-1}$, which means that the average of a large number of exponentially distributed values would be $\lambda^{-1}$.

The exponential distribution is characterized by its memoryless property. For example, if the first arrival has not happened for 1 minute, the probability that there will take more than an additional 30 seconds before the first arrival will be the same as the initial probability that it would take more than 30 seconds before the first arrival.

Another distribution which will be used later is the Erlang distribution. Similar to how the waiting times between two occurrences are exponentially distributed in a Poisson process, the waiting time between $k$ occurrences are Erlang distributed. The distribution is given by parameters: the shape $k$, and the rate $\lambda$. The expected value is given by $k/\lambda$.

### 2.4.2 Formulas

The formulas in this section are taken from the compendium from Enger & Grandell (2006). The service times, $\mu$, are stochastic variables which are independent and identically distributed. Stochastic variables are variables that can take on a set of different values with different probabilities. The service times are
also independent of the arrival process, \( \lambda \). Another variable is the number of service stations, denoted as \( c \). There are four different expected values which are of interest to this report. To understand these formulas, there are two variables which have to be introduced first. The first one is the traffic intensity, \( \rho \), which is defined as:

\[
\rho = \frac{\lambda}{c \mu}
\]  

(1)

For the formulas to work, \( \rho \) should be smaller than 1, otherwise the arrivals would completely overwhelm the rate at which customers are served in the system and the queue would grow to infinity.

The other variable is \( p_n \), which is the probability that there are \( n \) customers in the system. It is defined as:

\[
p_n = \begin{cases} 
\left( \sum_{l=0}^{c} \frac{(\rho)^l}{l!} \right)^{-1}, & n = 0 \\
n! \frac{(\rho)^n}{p_0}, & n = 1, 2, \ldots, c \\
n! \rho^n, & n = c + 1, c + 2, \ldots
\end{cases}
\]  

(2)

The formula contains three different cases which depend on the variable.

The first two of the four expected values are associated with the number of customers, and they are defined as:

- \( \ell_q = \frac{p_0 \rho}{(1-\rho)^2} \) expected number of customers in the queue

(3)

- \( \ell = c \rho + \ell_q \) expected number of customers in the system

(4)

The other two are associated with the time customers spend in the queue and system, they are defined as:

- \( w_q = \frac{\ell_q}{\lambda} \) expected time a customer spends in the queue

(5)

- \( w = \frac{\ell}{\lambda} \) expected time a customer spends in the system

(6)
3 Implementation

This project includes the development of a proof-of-concept prototype. The aim of the prototype is that it should be able to check-in a passenger with the help of NFC or a barcode. This chapter will describe the development of the prototype.

3.1 Hardware

- Google Nexus 4
- LG Optimus L5
- ACS ACR122U-A9 NFC reader

The two NFC enabled phones that will be used are LG Optimus L5 and Google Nexus 4. The Optimus L5 runs Android 4.0 and has a single-core CPU at 800MHz. The Nexus 4 runs Android 4.2.2 and has a quad-core CPU at 1.5GHz.

The ACR 122U is an NFC reader which can be connected to a computer via a USB-cable. It is compliant with both RFID and NFC standards.

3.1.1 Why Android

There are several reasons to why Android is the most advantageous platform for this project. At the moment, there are two major smartphone operating systems used in Sweden: Apple’s iOS and Google’s Android. In 2012, iOS was just ahead of Android with a market share of 48% compared to 36% for Android (Our Mobile Planet, 2012). However, Apple has yet to implement NFC technology into any of their yearly iPhone models (iPhone 5 being the most recent at time of writing), though their next model is rumored to be capable of NFC communication (Slivka, 2013). With a wide range of NFC enabled devices (Wikipedia, 2013) and a well-documented API (Android Developers, n.d.), Android is an ideal platform for this project.

3.2 Android development

This section covers explanations of the Android-specific classes Activity and Intent, in order to help the reader to better understand the prototype application. It also provides explanations for how to work with NFC and barcodes in Android.

3.2.1 Activity

Activities are described by Android Developers as “a single, focused thing that the user can do” (Android Developers, 2013). A user interface can be displayed inside an activity with layouts written in XML. An application usually consists of several activities and the user navigates the application with the help of an activity stack, which keeps track of previous activities. Activities has four different states.
• *Active or running* when the application is in the foreground.

• *Paused* when it is still visible but another non-full-sized or transparent activity is in the foreground.

• *Stopped* when the activity is not visible to the user, but still carries state and member information.

• A paused or stopped activity may be shut down by the system and must completely restart if it is to be displayed again.

### 3.2.2 Intent

Intents are described by Android Developers as “an abstract description of an operation to be performed” (Android Developers, 2013). One example of these operations is to launch an activity. There are different forms of intents; Explicit Intents and Implicit Intents. Explicit intents are normally used within an application to switch between activities; they explicitly state which class to run. Implicit intents do not explicitly state what to do; instead they must provide some information to the system to decide. The system uses Intent resolution to decide what to do with the implicit intents. Basically, activities may use an intent filter to specify which intents they want to handle and if the system receives a new intent which matches that filter, the activity will be started.

Data may be sent between activities as “extras”. Basically, when an intent is created, any data that the developer wants to send can be added to the intent as an “extra”. In the new activity, the data is received by retrieving any extra data that the intent carries. The API supports a wide range of data types for extra, such as int, string and boolean. Objects may be passed as extras if they implement either the Parcelable or the Serializable interface.

### 3.2.3 NFC

As stated in section 2.2.1 there are three operating modes for NFC devices. This section will provide information about which modes work in Android and how they are being used.

In reader/writer mode, the phone can read/write NDEF-messages (see section 2.2.1) from/to a tag. The tag can make the phone open up an application, for example opening up a web browser with the URL contained in the tag. This is done by a tag dispatch system that creates an intent with the NFC tag and sends it to any application that filters for that intent. To read a tag, the phone must be unlocked. The reason behind this is to make sure that the user intended to read the tag, a locked phone in a pocket may touch a tag but reading the tag would not be an intentional action by the user.

Peer-to-peer allows a user to send NDEF-messages to another device. The Android API uses the trademarked Android Beam for peer-to-peer connections. When using Android Beam, the application the user wants to send data from must be in the foreground, and after an NFC connection is established the user must then touch the screen to confirm the transmission. Android Beam is however not bidirectional,
which means that the two devices would have to re-establish the connection by touching again if the receiving device would like to send some data back.

With card emulation, the device interacts with a reader which is in reader/writer mode; the reader will read data from the device as if it were an NFC tag. However, while there is support for card emulation in the Android operating system (Google’s payment solution, Google Wallet, uses it), Google have opted for keeping the API hidden (Google Developers, 2012, 0:58:10).

### 3.2.4 Barcodes

Reading and creating barcodes on the Android can be done by using ZXing (ZXing, n.d.) and Barcode Scanner. ZXing is an open-source library for reading 1D and 2D barcodes; it is initially written in Java but has been ported to several other languages. Barcode Scanner is a very popular open-source app, which was written using ZXing.

### 3.3 Application

Two different methods were suggested. The first one would use the mobile application to communicate with a server to try and find a booking for the logged in user. NFC and barcodes would in this method be used to register the current location of the user.

The second method would provide a list of bookings made by the logged in user. The user would then transmit the booking number of a booking via NFC or a barcode.

#### 3.3.1 First attempt

The first method that was considered was to use the phone in reader/writer mode and interact with either an NFC tag or a QR code. The idea would be similar to the one used by the German Touch&Travel mentioned in section 2.3.3, by simply checking in by registering the location. There would be NFC tags and QR codes at the terminal containing information which identifies the location. The location along with a user identification and the current date and time is then sent to a server which looks for a booking made by the user, departing from the terminal at the given location and at close proximity to the date and time. If the server found a booking corresponding to the parameters it could send back a success message to the mobile phone and an error message if no booking was found. This method would therefore require an active internet connection during the check-in since it needs to communicate with a server.

This would be a solution that would be lightweight, easy to implement and quick to use. However, after inspecting the process at first hand, complications were found with this method. There will have to be some information transferred from the phone to check-in system in order to confirm the check-in and be able to print out a seating ticket to the user, which today is done verbally by telling the clerk your booking number or by having a printed booking document with a barcode. For example, after checking-in with the application, the ticket would have to be
verified. If simply a confirmation message or code would be shown to a clerk, it
would not be impossible to counterfeit and if a confirmation message were verbally
transferred it would not lead to any improvements over current processes. The best
way would be to electronically transfer a confirmation code from the phone, for
example via NFC or barcode, but then the initial check-in would be redundant
since nothing more than a booking number has to be transferred to perform the
check-in. It would also have been possible to make a copy of the QR code and
perform the check-in when not at the terminal.

3.3.2 Final prototype

After determining that the application has to be able to send information to the
check-in system there are two potential approaches; peer-to-peer and card
emulation. However, as stated earlier, card emulation is currently not supported by
the Android API, thus peer-to-peer over Android Beam will be the method used by
the application. Sending information over barcodes is more straightforward, since it
only requires the phone to generate a barcode which can be read by a barcode
reader from the phones screen.

3.3.2.1 Android application

NFC

In order to send messages via NFC, the application will need to support Android
Beam. The reader which will receive the message will be connected to a computer,
and thus the computer will need to run a program which can retrieve the message
and display its content. To use Android Beam from within an application, the
specific activity which the user wants to send data from must be in the foreground,
which means that each activity which wants to Beam data must implement their
own methods to do so. There are two methods available in the API to do this. The
first method is to simply call setNdefPushMessage with an NDEF-message in the
onCreate-method of the activity. The other method is to use the callback
setNdefPushMessageCallback, which lets the application create the NDEF message
when needed. The callback can be useful when the application should send some
dynamic data, for example the current time when sharing a Youtube-video. Since
the application will not need to send any dynamic data it will utilize the
setNdefPushMessage-method in onCreate.

NdefRecord[] records = {
    new NdefRecord(NdefRecord.TNF_MIME_MEDIA,
        "text/plain".getBytes(),
        new byte[0],
        Integer.toString(booking.getBookingNumber()).getBytes())
};
NdefMessage msg = new NdefMessage(records);
adapter.setNdefPushMessage(msg, this);

A record is created with the MIME-type “text/plain”, and the booking number is
converted to bytes and added as the payload. An NDEF-message is created with the
record and set to be the message to be sent in the setNdefPushMessage-method.

Barcode
The easiest solution to create barcodes is to use ZXing and its IntentIntegrator. A developer may create a barcode by creating an intent which will be handled by the IntentIntegrator.

```java
Intent intent = new Intent("com.google.zxing.client.android.ENCODE");
intent.putExtra("ENCODE_TYPE", "TEXT_TYPE");
intent.putExtra("ENCODE_DATA", data);
intent.putExtra("ENCODE_FORMAT", "QR_CODE");
startActivity(intent);
```

Basically an encode-intent is created and any parameters are passed as extras. The parameters used in the example are what kind of data to encode (text in this case), the data to encode (a text-string in this case) and what kind of format to use for the barcode. When the intent is started, it will be picked up by the Barcode Scanner application and a barcode will be displayed. A downside to this method is that Barcode Scanner has to be installed on the phone, if the application is not found then Google Play store will open up so the user may download the application.

### 3.3.2.2 Computer application

#### NFC

Retrieving the messages on a computer via a reader is a bit more difficult since there is no official library or method to use, but there are a few open source projects written in various languages available. The library that will be used is NFCTools which is written in Java. The reason for using this library is because it is known to work with the reader that will be used, ACR 122U, and since there is a collection of examples available as well as a small but active community, it makes working with the library easier. NFCTools supports two protocols for transferring data; SNEP and NPP. SNEP (see section 2.2.1) is defined by NFC Forum, and is primarily used by Android Beam. NPP (NDEF Push Protocol) is not an NFC Forum protocol but was used by Android before Android Beam was introduced in Android 4.0. Since both phones that are used are running on Android 4.0+, the application will be configured to use SNEP.

```java
NdefToSwing n2s = new NdefToSwing(nfcTextarea);
SnepServer ss = new SnepServer(n2s);
LlcpOverNfcip lon = new LlcpOverNfcip();
LlcpConnectionManager cm = lon.getConnectionManager();
cm.registerServiceAccessPoint(SnepConstants.SNEP_SERVICE_ADDRESS, ss);

TerminalHandler th = new TerminalHandler();
th.addTerminal(new AcsTerminal());
Terminal t = th.getAvailableTerminal();
NfcAdapter na = new NfcAdapter(t, TerminalMode.TARGET);
na.setNfcipConnectionListener(lon);
na.startListening();
```

The first row creates an object of a custom made class NdefToSwing, this class takes the output from the NDEF record and adds it to GUI. The terminal corresponds to the NFC reader, which is an ACS ACR122U-A9, and thus the class AcsTerminal is used. An NfcAdapter is then created with the terminal and the terminal-mode set to “target”, which means that the phone will initiate the
communication. If no ACS terminal is connected to the computer, the call to getAvailableTerminal() will throw an exception. The call to startListening() on the NfcAdapter will start a new thread in the supplied terminal, which will listen for incoming NDEF messages. When a NDEF message is received, it will be sent to a method in NdefToSwing and then appended to the GUI.

**Barcode**

For barcodes, a final product would use some sort of barcode scanner that is connected to the computer. The webcam on a computer is a much cheaper alternative and will suffice for the prototype. The application will use an open source Java library called Webcam Capture (Firyn, 2013) to enable webcam support in Java. ZXing will then be used to decode the barcode that will be captured by the webcam. This is the code to initialize the webcam:

```java
Dimension size = WebcamResolution.HVGA.getSize();
Webcam webcam = Webcam.getDefault();
webcam.setViewSize(size);
WebcamPanel panel = new WebcamPanel(webcam);
```

A resolution is first defined, in this case HVGA (480x320). The webcam is then retrieved and inserted into a WebcamPanel. The panel can then be added to the GUI.

To decode a barcode, the class implements the Runnable interface so the decode function is executed on a separated thread. This method utilizes a loop so it is constantly looking for a barcode to decode. The current image is captured with Webcam Capture, and ZXing is used to decode the image:

```java
BufferedImage image = webcam.getImage();
LuminanceSource source = new BufferedImageLuminanceSource(image);
BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));
Result result = new MultiFormatReader().decode(bitmap);
```
4 Results

This chapter will present the results of this project in three parts. It will start with a description of current processes at the Nynäshamn terminal and then move on to the conducted study at the terminal. The last part will present the results of the prototype.

4.1 Check-in process

The first visit to the terminal in Nynäshamn was on the 20th March 2013. The aim of the first visit was to examine current processes and routines by inspecting and talking to staff. The number of passengers was very limited, only about 50 on a ship which has a limit of 1500 passengers. The low congestion and non-stressful environment meant that it was easy to ask questions to the staff and observe the check-in process.

There are four check-in desks at the terminal, but only two were opened at the time of the visit due to the low number of passengers. The desks open 2h before departure during high season and 1h30m before departure during low season.

Most people have pre-paid tickets, but it is possible to buy tickets at the terminal (both at the check-in and at a separate desk), although at a higher price if the ticket is bought within two hours of departure.

When a passenger arrives at the check-in desks with a pre-paid ticket the preferred methods are either a printed copy of the ticket which can be scanned by the check-in clerk or a booking number. It is also possible to search for the passenger’s name, although this approach is a bit slower.

When the ticket is confirmed by the clerk and the passenger is marked as checked-in in the system, a seat ticket is printed. The seat number is automatically assigned by the system, but the clerks have the option to manually assign them as well, although it is a process which takes additional time and is thus a rare occurrence.

4.1.1 Discounts

There are a couple of discounts available and the requirements will sometimes be checked during the check-in. The most common discount that was checked during the visit was the discount for Gotlanders. After the ticket is found, the clerks can see if it is booked with the Gotlander discount (or any other discount, such as senior discount or student discount) and the clerk makes sure that the person is registered on Gotland by asking for the last four digits in their social security number (the first six are provided during the booking). If the social security number provided is not registered on Gotland, the passenger has to pay the difference at the check-in desk. When there are many passengers waiting in line, the clerks usually omit checking for discounts in order to save time and improve queue times.
4.1.2 Pre-check-in

Check-in for larger groups is a bit different. Since it is time consuming and not very practical to check-in a large group of people at the same time with the same ticket, it is possible to do a pre-check-in. Groups may pre-check-in before reaching the check-in desk to receive individual tickets which can be distributed by the group leader. Pre-check-in is available for larger groups or smaller groups where the individual passengers have to or want to check-in separately. Groups that want to do pre-check-in may do so at the desk which is outside the check-in hall. After doing a pre-check-in, each passenger in the group receives an individual boarding card, which is then scanned at the check-in desk to move the passenger from pre-checked-in to checked-in. The Gotlander discount, or any other discount, is not checked at the check-in desk when the passenger is pre-checked-in, although it may sometimes be checked during the pre-check-in. Figure 8 summarizes the check-in process.

![Diagram of the check-in process]

Figure 8 The check-in process

4.1.3 Time consuming activities during check-in

Not every passenger passes through the check-in without delay, during my two visits I took note of any delays that occurred and the following are based on those observations. The most common reason for passengers taking extra time at the check-in desk was passengers picking up their phones when they got to the check-in.
in desk and started looking for the booking number. This time consumer is pretty minor and does not hinder the flow of the process too much. Another common reason, which also took a considerable amount of time, was when passengers had made a re-booking or for some other reason had paid too much or too little and money had to be transferred at the check-in desk. Another time consumer is when passengers did not have either a booking number or document with a barcode so the clerk had to make a search on their name. Some passengers also wanted to check for alternative routes for their return trip and possibly make a re-booking. Another one, which did not occur during either visit, but apparently do happen from time to time according to the staff, is when a passenger with a Gotlander discount booking is not in fact registered on Gotland and starts to argue with the clerks over not wanting to pay the difference.

4.2 Survey

The second visit to the Nynäshamn terminal took place on Maundy Thursday 2013 when the ferries were almost fully booked due to the Easter holidays. The aim of this visit was to gather data to use with the queueing theory formulas. The complete results from the survey can be found in appendix A. The total amount of passengers was just short of the maximum capacity 1500, although most of them traveled with vehicles and only 412 passengers had regular tickets. I arrived at the terminal just before the check-in opened (two hours before departure), and sat at the regular check-in desks the whole time. The first hour was pretty calm and only a couple of passengers decided to check-in that early. After an hour though, the passengers arrived at a much higher rate, much due to the fact that the busses from Stockholm Central arrived at that time. In the end 406 passengers had checked-in at the regular check-in desks.

Times were measured by sitting at the check-in desk and starting a timer when a passenger arrived at the desk and stopping the timer when they received their tickets. The passenger was considered to have “arrived” at the desk during the greeting between the clerk and the passenger. Bookings for several passengers were generally quicker per passenger. Although it took a bit longer with more people, since more seat tickets had to be printed, there is some overhead which does not increase when passengers in the group increases and therefore times for groups tend to be a bit lower per passenger.

In the end, I managed to get time measures on 42 check-ins (71 passengers). It was limited due to the four check-in desks meant that roughly 25% of the passengers passed through the one I was sitting in. I also only managed to time roughly every second passenger due to having to write the time and any notes down before beginning to time the next passenger.

The average check-in time was 20.82 seconds (measured on 34 check-ins) when there were no extra time consuming activities during the check-in. The average check-in time when taking extra time consuming activities into account was 35.1 seconds, which was measured on 42 passengers (which means 8/42 passengers or almost 20% needed extra service time from the clerks).

23
The most common extra time consuming activity was passengers that had paid too little or too much for their tickets. This happened 5 times, which means 62.5% of all time consuming activities or almost 12% of all check-ins.

### 4.2.1 Calculations on current processes

The formulas in section 2.4.2 require three parameters, the number of service desks, the arrival rate and the service time. The maximum number of service desks at the terminal is four, and all four were opened during the survey. The average service time was 35.1 seconds, as mentioned in section 4.2.

After the check-in had closed, 406 passengers had checked-in at the desks, however the number of bookings are of more interest than the number of passengers since if two or more passengers have the same booking, they will queue together and be served by the clerk together. There were a total of 71 passengers checked-in in my survey and 42 bookings, which means that on average, each booking had 1.69 passengers. Using this ratio, out of the total number of 406 passengers, there is estimated to have been 240 bookings. The arrival rate could be said to be the total number of check-ins over the time the check-in was opened (240/2h), however since the arrivals were so irregular this would not give a very good representation of the actual process. Very few passengers arrived during the first hour that the check-in was open and almost every passenger had checked-in 15 minutes before departure. This paper will utilize an estimation that 90% of the passengers arrived during the 45 minute period, which begins 1 hour before departure until 15 minutes before departure. This means an arrival rate of 216 check-ins/45 minutes, or 4.8/minute. Table 2 displays the result of the calculations with these numbers.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$, arrival rate</td>
<td>4.8/minute</td>
</tr>
<tr>
<td>$\mu$, service rate</td>
<td>1/35.1 seconds</td>
</tr>
<tr>
<td>$c$, number of desks</td>
<td>4</td>
</tr>
<tr>
<td>$\rho$, traffic intensity</td>
<td>0.7024</td>
</tr>
<tr>
<td>Average number of customers in the system</td>
<td>3.8301</td>
</tr>
<tr>
<td>Average number of customers in the queue</td>
<td>1.0205</td>
</tr>
<tr>
<td>Average time a customer spends in the system</td>
<td>47.8427 seconds</td>
</tr>
<tr>
<td>Average time a customer spends in the queue</td>
<td>12.7474 seconds</td>
</tr>
</tbody>
</table>

Table 2 Queue theory calculations
It is important to note that these numbers just gives an estimation. In reality, the arrivals came in bursts, due to passengers arrived with busses and trains, so they were not as randomly separated as assumed in the Poisson process.

Another performance indicator for the system is how fast all customers can be served. A worst case scenario would be if all passengers would arrive at the same time, which is not far from the truth since most passengers arrive at roughly the same time with the busses. To estimate how fast the customers could be served if they arrived at the same time, the Erlang distribution, mentioned in 2.4.1, will be used. If the service times for each passenger are exponentially distributed, then the service time for k passengers are Erlang distributed. The average time it will take to check-in all passengers will be given by the expected value of the Erlang distribution. As mentioned above, the total number of bookings to check-in is 240 and the service rate is 1/35.1 seconds. The expected value will thus be 240*35.1. Since there are four service stations to handle the check-ins, the expected time can be divided by four, which means that the expected time to check-in all passengers will be 2106 seconds (approximately 35 minutes).

### 4.3 Application

The solution will make use of a self-service check-in station which will be available for certain passengers. Since some passengers require interaction with a clerk (re-bookings, payments etc.) they cannot use the self-service check-in station and will have to go to a manned check-in desk. A notification will be sent out a pre-determined time before departure, to the passengers who can use the self-service check-in. When these passengers send their booking number over NFC or with a barcode, the system will recognize their booking as qualified for the self-service check-in, check them in, print out a seating ticket and unlock a turnstile they can pass through. Gotlander discounts can be set to allow everyone (as it is today when there are many passengers) or spot-checking passengers by requiring random passengers to scan the barcode on their ID-cards.
Figure 9 List of bookings made available after logging in

Figure 10 View after selecting a booking from the list

Figure 9 is the view the user sees after logging in or after starting the app and already logged in. It shows a list of all the bookings belonging to the logged in user. The user can select a booking in the list to get a view of the selected booking, as depicted in Figure 10. The screenshot shows a text saying to touch the phone to an NFC reader in order to transmit the booking number, if the phone does not support NFC or if NFC is disabled in the settings it will instead display a text which notifies the user about it.
Figure 11 shows the view the user will see when they click the button “Show barcode”. Figure 12 shows the notification which will be shown to the user a couple of hours before departure. If the booking is not fully paid yet, there will be a message telling the user to use one of the regular check-in desks instead.
Results

Figure 13: The desktop application at startup with barcode data from webcam in top-right and data from NFC in bottom-right.

Figure 13 is a screenshot of the desktop program just after it is launched. The left panel shows the webcam view and the two panels to the right will contain text with the data read from the phone. The upper-right panel shows what is read from barcodes by the webcam, and it will display the text “Webcam ready” if there is a webcam available on the computer. The lower-right panel shows the data that is read by the NFC reader from the phone, and if there is a reader connected to the computer, it will display the text “NFC ready”.

Figure 14: Desktop application reading a QR code.

Figure 14 is a screenshot of the desktop program when it reads a barcode from the phone. The webcam panel to the left shows the phone screen with the barcode which is displayed to the webcam. The booking number is correctly displayed in the upper-right panel.
Figure 15 Phone communicating with reader

Figure 15 is a photo when the phone is making contact with the NFC reader. The Android Beam interface pops up and prompts the user to touch the screen to send the data.

Figure 16 Booking number transferred via NFC to the desktop application

Figure 16 is a screenshot of the desktop program after it has received data from the NFC reader. The booking number is correctly displayed in the lower-right panel.
4.4 Calculations for proposed system

Since the terminal only has room for four check-in desks, one of the current desks has to be changed to a self-service desk, which means that there will be one self-service desk and three regular check-in desks. The first matter which needs to be settled is which passengers that are eligible for a self-service check-in. Passengers with no incoming or outgoing payments should be eligible, as well as passengers with no additional matters, like making a return booking. This would make 34 out of 42 of the bookings in the survey eligible, or 81%.

There will be 3 manned check-in desks with 19% of the arrivals and one self-service check-in with 81% of the arrivals. The service rate for the manned desks is an average of the 8 check-ins in the survey which would not be eligible for self-service check-in. The service rate for the self-service check-in is estimated from the application. Sending the booking number over barcode or NFC could be done in less than 5 seconds, even in the cases when the number was not successfully sent at the first attempt. A few seconds for registering the check-in and printing out a seating ticket will have to be added and thus a service time of 8 seconds will be used for the self-service check-in.

<table>
<thead>
<tr>
<th></th>
<th>Current manned system</th>
<th>Manned desks</th>
<th>Self-service desk</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$, arrival rate</td>
<td>4.8/minute</td>
<td>0.915/minute</td>
<td>3.888/minute</td>
</tr>
<tr>
<td>$\mu$, service rate</td>
<td>1/35.1 seconds</td>
<td>1/95.75 seconds</td>
<td>1/8 seconds</td>
</tr>
<tr>
<td>$c$, number of desks</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$\rho$, traffic intensity</td>
<td>0.7024</td>
<td>0.4867</td>
<td>0.5185</td>
</tr>
<tr>
<td>Average number of customers in the system</td>
<td>3.8301</td>
<td>1.6713</td>
<td>1.0767</td>
</tr>
<tr>
<td>Average number of customers in the queue</td>
<td>1.0205</td>
<td>0.2112</td>
<td>0.5582</td>
</tr>
<tr>
<td>Average time a customer spends in the system</td>
<td>47.8427 seconds</td>
<td>109.6017 seconds</td>
<td>16.6134 seconds</td>
</tr>
<tr>
<td>Average time a customer spends in the queue</td>
<td>12.7474 seconds</td>
<td>13.8517 seconds</td>
<td>8.6134 seconds</td>
</tr>
</tbody>
</table>

Table 3 Calculations with self-service check-in
Table 3 contains the calculations for the two different types of desks when using a self-service check-in. The average passenger will have a queue time of 9.6 seconds (81% of the passengers have an expected queue time of 8.6134 seconds and 19% of the passengers have an expected queue time of 13.8517 seconds). The calculations from Table 2 are included in the first column for comparison purposes. Note that while the average time a customer spends in the system is much longer at the manned desks, the time they spent in the queue to the manned desks is essentially the same as for the current manned system. The difference between times in the system is because the service time is longer than before, so while the average customer spends the same time in the queue they will spend more time at the check-in desk.

Since the service time is an estimated value, it can be interesting to look at how the service time affects the queue time, as can be seen in Figure 17.

![Queue time graph](image)

**Figure 17** Queue times for different service times

As the service times increases, the service rate will be closer to the arrival rate and the traffic intensity will increase. As can be seen in formula 3 in section 2.4.2, the number of customers in the queue (which affects queue time) grows exponentially as the traffic intensity increases. A service time of 9 seconds gives a queue time of roughly the same as it was with 4 check-in desks.

When looking at how fast all passengers would be checked-in if they all arrived at the same time, the queues for the self-service check-in and the manned check-in will be calculated separately. The self-service check-in will have 81% of the 240 check-ins and has a service rate of 1/8 seconds. The expected time to check-in all passengers is thus 1555.2 seconds (approximately 26 minutes) for the self-service check-in. The manned check-in will have 19% of the 240 check-ins and has a service rate of 1/95.75 seconds. With three check-in desks the expected time to check-in all passengers at the manned desks is 1455.4 seconds (approximately 24 minutes).
5 Discussion

This chapter will discuss the advantages of the developed prototype from two perspectives. The first part will discuss any advantages to the service time, in other words how the prototype can speed up the check-in process. The second part will discuss other improvements, such as how the prototype encourages the passengers to travel paperless and advantages in doing so. It will also present the error sources.

5.1 Service time

One of the most common delays during the check-in process was when the passenger had to make extra payments or had paid too much and had to be paid back money. Reducing the number of passengers who need to make payments at the check-in desks would significantly improve the process. One alternative could be to push out messages (for example by SMS, email or via the application) to the passengers with incoming or outgoing payments and allow them to pay beforehand, e.g. by sending an invoice or by credit card. This notification could either use the application to perform payments, or it could provide a link which they can open in their browser. When using invoices, the application or website could ask the user to input a social security number to accept to be billed with the specified amount. When using a debit or credit card, the application or website could implement some payment solution such as PayPal or DIBS to handle the transaction. When the passengers pay or accept to be billed, their booking will be marked as eligible for the self-service check-in.

Receiving a notification will raise awareness for the application and work as an incentive to its usage. A notification to an eligible passenger makes sure that the feature is known to the user and will also work as a reminder to passengers who are already aware of it. A notification to a non-eligible passenger may encourage them to fully pay their tickets ahead of arriving at the terminal, and thus reduce queue times for passengers using the manned check-in desks.

As stated in 4.4, the time an average customer spends in the system at the manned desks is longer than before; however the time they spend in the queue is roughly the same. This is due to the fact that the average service time is much longer. Since the service time directly affects the time spent in the system, the time spent in the queue would be a better performance indicator. For example, even if the queue time would be 0 for the case with 3 manned check-in desks, the total time spent in the system would be longer than before. When looking at the other factors, they all indicate a better performance - the traffic intensity is smaller, which means that the system is further away from it maximal capacity. The number of customers in the system and the queues are also smaller.

The self-service desk also performs really well, having the lowest queue time. It is important to note that a service time of 8 seconds is an estimated value, based on the fact that sending the booking number from the phone takes around 3-6 seconds (slightly higher when using barcodes), and a few seconds are added to let the
system perform the check-in and print out tickets. Altering this number by just one second can affect the queue time with several seconds. An average service time of 9 seconds would be the absolute maximum since it would give a queue time which would be the same as before. However, after evaluating the application, my assessment is that an average service time should be lower than 9 seconds.

The time it takes to check-in all passengers if they would arrive at the same time is another performance test where the new system comes out on top. Both the queue for manned check-ins and the queues self-service check-in are processed quicker compared to current processes, with a decrease from roughly 35 minutes to 26 minutes. In this model, the queues are naturally longer than if the arrivals would be modeled by the Poisson process. The longer queues give a better representation of how the passengers arrive at the terminal, since a majority arrives at the same time with the busses. No matter how the arrivals were modeled, the new proposed system was faster.

The introduction of a notification/reminder for the passengers to fully pay their tickets before arriving at the terminal will lead to a higher percentage of passengers being able to use the self-service check-in. Having too many passengers passing through the self-service check-in could potentially cause congestion and it could be beneficial to eventually open up more self-service check-in desks.

It is difficult to say if the introduction of NFC and QR-codes or the new proposed system with a self-service check-in is the dominant factor in improving the queue times. While NFC and QR-codes managed to improve the service times when checking-in, the proposed system will improve the throughput for the passengers eligible for the self-service check-in as well as reduce the number of passengers in the queues to the manned check-in desks.

5.2 Other improvements

Besides the improved service times for passengers using the self-service check-in, there are other improvements the proposed system would bring. Most importantly, it provides easier more convenient options for the passengers to travel paperless. Advantages of traveling paperless are that passengers do not have to think about the hassle of printing out a ticket and remembering to bring it with them when traveling. Having it on the phone also reduces the risk of losing the ticket, since most people usually have their phones close at hand, but a paper ticket can easily be misplaced or forgotten. Paperless travelling is also more environmentally friendly than using paper tickets.

It is possible to travel paperless with Destination Gotland today when checking-in with a booking number. Check-in with a booking number does however not necessarily mean that the passenger travels paperless, some passengers come with the number written down on a piece of paper. Retrieving the number when stored on the phone is not performed in a uniform way and is not necessarily quick and easy. Most passengers open up their email application to look for the booking number in the booking email, which can take some time if there are many emails in the inbox. Another disadvantage to checking-in with the booking number is that the
passenger is told to read the number to the clerk who types it in. While it is only marginally slower than scanning a barcode, it can be error prone. The passenger may read the number wrong, for example by leaving a number out or by accidentally repeating one, or the clerk may enter the wrong number into the system. These errors are not very big, they can easily be rectified by redoing the process and in case the passenger has written down the wrong number, the clerk can search for their name instead, however if an error occurs it will take extra time and slow down the process. When transferring the booking number electronically through a barcode or NFC, granted that the hardware actually works and the data is transferred, the booking number is guaranteed to be the correct one.

As mentioned in the previous paragraph, the current way of retrieving the booking number from the phone is not necessarily a smooth and convenient operation. The application that was developed tries to make this more convenient for the user since it stores the booking information somewhere where it is easy to access for the user. The application displays a list of the bookings made by the logged in user where they can select a booking to send its booking number via NFC or generate a barcode of the booking number so it can be read by a scanner at the check-in desk.

### 5.3 Sources of error

The first error source is that the values used for the queue theory calculations are based on one visit to the Nynäshamn terminal. The service times would probably be similar on different occasions, although a bigger data set than 42 would provide a more accurate value for the service time. The arrivals however, could vary quite a bit. During my visit to the terminal, there were a large number of vehicles on the ferry, which is usually the case. It is however important to point out that more vehicles means a smaller arrival rate at the regular check-in and fewer vehicles means a larger arrival rate at the regular check-in.

There are also a couple of estimations which impacts the results. The first one is the estimation for the arrival rates. Since the arrivals are so spread out, with only a few check-ins during the first hour, using the whole time period of two hours would not work well with exponentially distributed arrivals. The estimation to work with a smaller time period during which the arrivals were more frequent is a fairly rough estimation.

The other estimation was made for the service time at the check-in desk. The prototype was never tested with a real check-in, so the service times were estimated based on how long time it took to send the booking number from the phone to the computer. The number used for the service time is just off the “break-even”, when the queue time would not be improved, and an increased average service time of just a few seconds would impact the queue time immensely. However, it is important to note that the calculations are based on the fact that the percentage of passengers that are able to use the self-service check-in are based on the number during the one visit to the terminal and that all passengers that can use the self-service check-in are in fact using the self-service check-in. In a real-life implementation, there probably are passengers that would not want to or for
technical reasons would not be able to use the self-service check-in and would thus have to use the regular check-in desks.

5.4 Conclusions

In conclusion, the project was a success. While the prototype was never integrated with the booking software or tested in a real-life situation, it achieved its intended purpose by serving as a proof-of-concept of sending a booking number via NFC or barcode to a PC. The prototype also managed to show that the proposed system, with mobile check-in, would, at least in theory, be able to reduce queue times at the terminals. Since the data used in the survey was relatively small, more surveys with a bigger data set could be performed for more conclusive results.

Apart from the queue times, there were other improvements which were beneficial not only for the customers but also for the employees at the terminal. For example, notify passengers who have to pay for their tickets at the terminal and encourage them to pay ahead, can help decrease the workload of the check-in staff.

While there is still a great deal of work to be done before it can be used in real-life situations, mainly integrating with the booking system, there are many improvements over the current processes which could make it worthwhile.
References


References

Available at: http://en.wikipedia.org/wiki/List_of_NFC-enabled_mobile_devices
[Accessed 04 04 2013].

Napa: Westsong.

ZXing, n.d. zxing: Multi-format 1D/2D barcode image processing library with
clients for Android, Java. [Online]
Available at: https://code.google.com/p/zxing/
[Accessed 04 04 2013].
## Appendices

### A Survey

<table>
<thead>
<tr>
<th>Time (seconds)</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>2 persons</td>
</tr>
<tr>
<td>147</td>
<td>Cash payment, name search, check return departures, change seating</td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>Extra payment</td>
</tr>
<tr>
<td>65</td>
<td>Repayment</td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>3 persons</td>
</tr>
<tr>
<td>20</td>
<td>2 persons</td>
</tr>
<tr>
<td>109</td>
<td>Name search, re-booking of return trip</td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>3 persons</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>28</td>
<td>2 persons</td>
</tr>
<tr>
<td>56</td>
<td>2 persons, extra payment</td>
</tr>
<tr>
<td>14</td>
<td>2 persons</td>
</tr>
<tr>
<td>26</td>
<td>2 persons</td>
</tr>
<tr>
<td>150</td>
<td>Name search, could not find due to wrong name on ticket</td>
</tr>
<tr>
<td>124</td>
<td>Repayment, debit card problems</td>
</tr>
<tr>
<td>50</td>
<td>Extra payment</td>
</tr>
<tr>
<td>32</td>
<td>4 persons</td>
</tr>
<tr>
<td>18</td>
<td>2 persons</td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>4 persons</td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>2 persons</td>
</tr>
<tr>
<td>20</td>
<td>2 persons</td>
</tr>
<tr>
<td>18</td>
<td>2 persons</td>
</tr>
<tr>
<td>21</td>
<td>3 persons</td>
</tr>
<tr>
<td>36</td>
<td>4 persons</td>
</tr>
<tr>
<td>18</td>
<td>2 persons</td>
</tr>
<tr>
<td>15</td>
<td>3 persons</td>
</tr>
<tr>
<td>16</td>
<td>2 persons</td>
</tr>
</tbody>
</table>

**Table 4** Result from survey