Mobile Application for Informative Communication During Sports Events

Teodor Carlsson
Linnea Gunnarsson
Sebastian Lindgren
Maria Lindqvist
Abstract

Mobile Application for Informative Communication During Sports Events

Teodor Carlsson  
Linnea Gunnarsson  
Sebastian Lindgren  
Maria Lindqvist

The student sports event Studentiaden is a yearly occurring event arranged by Swedish University Sports Federation (Sveriges Akademiska Idrottsförbund) in collaboration with local student sports associations. During Studentiaden approximately 500 to 1000 students participate every year from all across Sweden. The size of the sports event makes it difficult to reach out to all competitors and visitors. In this project, a mobile application was developed to allow faster communication from the organizers to the competitors and visitors with the use of push notifications. The application consists of relevant information regarding the event, such as schedules, results and contact information. A stress test with 1500 connections simultaneously was made which the application passed. Another test run on the mobile application was the navigation time to different pages in the application. The results attained from all tests were successful. The application was never published due to expected long waiting times for registration at Google Play and App store. However the application as it was provided an alternative means to transfer information to participants and visitors from the arrangers at Studentiaden.
Sammanfattning

# Contents

1 Introduction 1

2 Background 1

3 Purpose, Aims, and Motivation 3
   3.1 Delimitations 4

4 Related Work 5
   4.1 Other Applications 5
   4.2 Other Ways to Communicate 6
   4.3 Notifications for Websites 6

5 Method 6
   5.1 Programming Languages and Environment 7
   5.2 Libraries 8
   5.3 Choice of Database 9
   5.4 Choice of Server Host 9

6 System Structure 10

7 Requirements and Evaluation Methods 12
   7.1 Robustness 13
   7.2 Customizable Content 13
   7.3 Usability 14
   7.4 Publication Guidelines 14

8 Database Structure and Organization of Data 14
1 Introduction

How do you most efficiently reach out with important information to thousands of people during an event? There are many different ways to reach out to competitors and visitors (from now on referred to only as participants). During an event such as a sports event, there is a lot of information the participants might need. Relevant information could be schedules, results, maps and updates about last minute changes. To distribute this information the organizers could update a web page, making speaker announcements or setting up posters. A different approach is using a mobile application for smartphones in which participants can interact.

In this project, a mobile application was created for a sports event for students. The purpose was to give the participants an option for quickly finding relevant up-to-date information about the sports event. This would provide both high portability and easy distribution of updates, since most people tend to carry a mobile phone with them. By using the mobile application, participants of the sports event can access the information they need, all in one place. Using a mobile application is advantageous for its availability, since most people today are carrying a smartphone on a daily basis [KA16]. The mobile application was considered a more modern choice that could replace the need to look for posters or stationary displays which may be limited to one or a few locations.

This particular mobile application was specifically made for Studentiaden, which is an annual student sports event in Sweden. Since the mobile application was made to be adjustable, it can be customized for each year. The application will be made in two different versions. One version will be used by the participants, which as mentioned are both visitors and competitors. The other version will be used by the organizer group. These two groups will in this report also be referred to as users and administrators respectively.

The mobile application was developed in React Native which is a framework for creating native mobile applications for Android and iOS smartphones using the programming language JavaScript and the React library. Further explanation on React Native can be found in section 5.1.

2 Background

Studentiaden is an annual sports event organized by Sveriges Akademiska Idrottsförbund, also known as Swedish University Sports Federation. Studentiaden is co-arranged with local sports associations from the city where the event is being held. The event is held
in different cities throughout Sweden each year. In 2018, Studentiaden will be held in Karlstad and co-arranged by Karlstad University Student Association of Sports.

The participants of Studentiaden are students from various universities in Sweden who are members of a student sports association. A participant of Studentiaden can compete in any of the arranged sports, given that the team meet the requirements regarding the number of team members. The arranging local sports association decides which sports are available for the competition. Among these, four sports are Swedish championships for students, while four sports are not. Instead, these will be called tournaments. This was to encourage more students to participate by not calling it a championship as well as allowing students without long practice or experience to take part. During the event, there will also be a reward for the club with the highest total score. Clubs get score for medals won in games, as well as for number of participants in tournaments to encourage participation.

In events like Studentiaden there is a need to get out updated information to the participants. Personal schedules, schedules for each sport, results and maps over the area are examples of information that may be needed before or during the event. According to Studentiaden's organizers, this information is currently found on posters and their website.

The problem with posters and websites is that if the information is changed, the information may need to be constantly updated so that outdated information does not misinform or the latest information is missed altogether. The updated information would ideally be distributed to as many participants as fast as possible. Not doing so may result in confusion or frustration for the participants. Although it might be convenient to change information on a website, a website is not always easily accessible on the move. Using posters may also create inconvenience for the participants since they would have to physically find a posting board. Specifically in the case of Studentiaden, posters might be located in different spots across the sports halls. This will lead to only having access to this information when an individual pass by such a spot. An urgent update such as a change in schedule might not be delivered to the participants on time. A mobile application however would be able to bypass these problem for people with a smartphone. In 2016, 81 percent of the Swedish population owned a smartphone and the numbers are still increasing [KA16]. Therefore it is safe to assume that the absolute majority of the participants owns a smartphone, making a smartphone application a reasonable approach.

One other method of distributing information is by sending out emails. When using email as a means of updating participants with information, there is a high risk that the messages are not read until long after they have been sent. They may also be ignored or filtered out. There is also a risk that the email is being hidden amongst the many other
emails that are received. One concept that describes this effect is “email overload” [FBGS], which is the state where email is used for much more than communication. This in turn could lead to a cluttered e-mail inbox and loss of information.

Using social media, it is not for certain that all participants are using one particular type of social media. Assuming so would leave out those that do not use that particular social media. One approach to the problem is to develop a mobile application where all necessary information is available regardless of the users physical location, instead of using social media. A mobile application can also help the organizers to more easily send out messages and make them accessible to the participants without being dependent on a specific location such as a notice board.

Our stakeholder is Swedish University Sports Federation, the organizer of Studentiaden, for whom the mobile application was developed. They connect the majority of the student sports associations in Sweden. The federation is also open for students that are not associated to any sports association but wish to participate in sports. They arrange events and education for the associations and represents these in student sports issues on both national and international level. One of their goal is to make sports available to every student in Sweden no matter their current skill level. There are approximately 60 000 members and almost 100 connected sports associations which together accommodate 50 different sports [Sve18].

3 Purpose, Aims, and Motivation

The main purpose of the project was to enhance the overall experience for participants of Studentiaden. To do so, a mobile application was developed that gives the participants a way to keep track of what is going on and a way to find useful information. It also provides a way for organizers to distribute updated information to the participants.

The aim of the project was to provide a technical solution for Studentiaden, as well as for future events with similar problems and needs. The application combines schedules with sport results, amongst other features. This was done in hopes of creating an application that was beneficial for the individual participants of Studentiaden. The aim was also to work towards making the application user friendly with a design that was easy to navigate through while also making the system robust enough so that it can handle many requests at the same time.

The motivation for the project was the need of an easy to use platform for centralized information, that would be easy to access by all participants of Studentiaden. With the use of push notifications, information in the form of messages can be sent out to the
users of the application when needed.

One other issue of Studentiaden was the fact that the event is organized in different locations each year. This means that there may be a lack of knowledge regarding the city in which the event takes place, since participants comes from all around the country to attend the event. This increases the need of information about the local area. An attempt of filling this increased need is done by integrating an interactable map that has both information about the sports arenas locations as well as other common places in the surrounding area such as supermarkets and train stations.

### 3.1 Delimitations

In the early stages of the project, an idea was to enable the creation of personal accounts in the application for the users. These accounts would be used for personalized schedules, chatting with other users or showing support towards the playing teams in ongoing matches. These accounts could also be associated with the registration process on the Studentiaden’s website. However this would likely require access to possibly sensitive personal information. The new European law, GDPR, for storing personal information [IT-18] made it secondary to implement the feature due to the time restriction of the project.

It may also be convenient for the users to be able to mark events that are interesting to them. By marking the events, they would be saved locally and show up in the personal schedule. Storing self customized individual information was never implemented. Instead focus was primarily towards more general information that is useful for all users and not only for individuals.

Another desirable functionality of the application was to have something that would resemble cheering on an ongoing match, as requested as an optional feature by our stakeholder. The purpose of that would be to allow cheering on teams or associations during the ongoing games, even if the user are not physically present. However, it was decided to be an extra feature that was not prioritized.

There was also indication of a desired chat function for the application by our stakeholder. However, this was deemed to take a considerable amount of time to apply proper security aspects and was therefore considered a low prioritized matter. A chat would also be slightly out of scope for our project and was therefore never implemented.

The application was not developed for any operating system other than the two most commonly used ones, Android and iOS, since those two alone cover an overwhelmingly majority of smartphone users [KA16].
4 Related Work

In this section, related work is presented. The first part will cover other applications that provide information and communication during events. The second part present an alternative way to communicate. The third part describes how push notifications for websites are used and how they work.

4.1 Other Applications

There exists a few similar mobile applications that are built for events, like EventBoard [eve18], WebMobi [Web18b] and the Olympic Games application [Int18]. These are usually made by big companies and offered to event organizers with a slightly different design for each event. The applications often consist of schedules, lists of other participants of the event, chat rooms and push notifications. They are often linked to Instagram, Facebook and/or Twitter to make sharing moments easy.

The most similar mobile application to ours was the official application for the Olympic Games. That application consists of schedules and result lists for the different sports, numbers of medals won in each sport, each country and in total. It contains information about the different countries and about the sports. It also provides information regarding visiting the event, like nearby events, how to get to places by public transport or car, and accommodations.

Our application was similar to the Olympic Games application in certain aspects but differs in others. Our application was, as opposed to the Olympic Games application, developed for smaller events. This makes it possible to customize the application for more niche target audiences. It was also more custom designable in the sense that it could be used as a foundation that could be reused and modified for different events. The modifications in turn could be themes for different events, colors of text as well as icons for different events, and of course the information itself.

The map in the application includes the sports event location, nearest supermarkets and convenient stores, key locations in the area that may be of use to the participants of the event. It will not show locations that may not be as useful for this particular audience such as expensive restaurants or other locations that may not be much relevant for students. The application also shows competitions won by association. Moreover it is possible to handle the top-list for the associations, which includes giving points to associations for participation and victories in the games. Finally, push notifications was also implemented to allow last minute changes which might be needed for smaller events, something that the Olympic Games application does not have.
4.2 Other Ways to Communicate

The concept of transferring information from one medium to another, in our case by transferring information from analog media such as posters to digital media on a smartphone, is in some ways related to a work that was done by a group at the University of Minho [RJ13]. They worked with displaying information on displays located in public areas. This information was generated by having people checking in at geographical locations. They did this in order to poll what different teams people where supporting and visualized this on the display. Our project is similar in the sense of moving away from posters and instead developing an alternative way to distribute information. What was interesting about this was that the displays, although stationary, could update information once and all monitors would show the updated information. In our case, this could prove to be useful since games are being held at different halls simultaneously.

4.3 Notifications for Websites

Push and notifications are two different actions that cooperate to allow websites to interact with the user. Push is the action when a server is sending information to a service worker. A service worker [Mat18b] is a script that runs in the background of the website to allow features that do not need a web page or that does not interact with the user. After the information is sent to the service worker the service worker itself sends the information to the website, displaying a notification. The notification is showed on the web page and the user can interact with it [Mat18a]. This is similar to how push notifications for smartphones work, which is further described in section 9.

Web notifications are used by several different websites like social networks, news pages or blogs. They are used for the same reasons as push notifications for smartphones, to keep users updated about news, for advertising, to keep the user interested and keep them coming back.

5 Method

In this section, the methods and tools that were used during the project will be discussed. First is a description of the chosen programming language JavaScript as well as a comparison to other possible programming languages for application development. Thereafter it is discussed which libraries were used and our choices concerning the database and cloud service provider.
5 Method

5.1 Programming Languages and Environment

One of our requirements was that the application should be able to run on both iOS and Android devices. When writing applications for Android, Java is the most used programming language. Android phones are also supporting C/C++ [And18a] and Kotlin [And18c] programming languages. Since Android applications are primarily made using Java, it also comes with plenty of libraries that makes communicating with the phone easier. These libraries can be for actions like alerts, push notifications and user interface components that are easy to import and use. For iOS applications the programming languages used are Objective C, C or Swift. Swift is built upon C and was developed by Apple Inc [App18c]. Our option of choice regarding the programming language was React Native, a library for JavaScript which works for iOS and Android. However for the iOS platform there are some actions that are not available in React Native. The problem of missing actions can be solved though since React Native can communicate with Objective C and Swift.

Writing natively was another option, that means writing code for Android and iOS separately. This essentially means that one would have to write two separate versions of the same application. In our case it would mean to write one application for Android phones in Java and one application in Objective C for iOS. Doing so would mean having to write the same code twice, with a few variations here and there specifically for each operating system. A big problem with this is that it is not time efficient and difficult to get the applications to both look, feel and work the same.

JavaScript is an object oriented language more commonly referred to as a scripting language [Chr18]. JavaScript is mostly used on the client side of building websites but can also be used as the base language for Nodejs. Nodejs is used to handle the server and manage all different requests and inputs to the server [Nod18].

An alternative would have been to create a web application and wrap into a mobile application. This means that a web application written in HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript will run as a mobile application. An advantage of doing so would be the ability to write as if it would have been a website. Doing so would also allow a cross-platform tool such as Codename One [Cod18] or PhoneGap [Ado18] instead of writing for Android and iOS individually in their respective programming environments. However writing a web application in this way could come at a cost of performance [HHM13]. There will be overhead costs for wrapping the web application into an application that runs on the phone since it is not written in native code and must make translations which takes time.

React Native was developed by Facebook engineers and publicly released in 2015 and is getting more and more used in web development [Bra18]. It works as a library on top
of JavaScript and can be used to create the user interface of a mobile application. It uses two separate threads that communicate through a bridge [Bla17]. The main thread is running in every mobile application and handles the interaction with the user. The other thread runs the JavaScript code with an JavaScript engine and are exclusive for React Native applications. The JavaScript handles the logic of the application. The bridge makes sure that the two threads never block each other and never share the same data. It also makes sure that shared messages between the threads is transferred in an optimized way. This allows React Native to communicate with the native languages of Android and iOS.

Because of this, the application was developed with React Native. By using React Native it was possible to use code written once for both iOS and Android. The great advantage of this is that the code can be changed at one place instead of having to adjust code for two separate platforms. Another advantage was also the fact that it was not necessary to know two different languages, one for each operating system. Not having to customize over two different languages saves a lot of time while developing the application.

5.2 Libraries

Several JavaScript libraries were used in order to develop the user interface as well as the backend functionality. JavaScript and React Native is used to develop the user interface which is described in section 11.

For the application to communicate and make requests for data from the database on the server the GraphQL [Fac18] library was used. GrahpQL handled the communication between the client and server and set up the connection as well as sending and taking care of the queries which the application sent. One of the advantages of GraphQL is that it allows for clients such as the application to ask for the data it need from the database and get exactly that information back with minimal amount of requests to the server [Phi18]. This is especially good for mobile applications since there will be less traffic on the network because fewer requests are made, which saves battery [App18a] and network data usage. GraphQL also works as a query language interface that allows the client to only use predefined queries that the server can take care of and nothing else. This means the server knows what queries to expect and the client cannot create own queries. The actual fetching of data will be the GraphQL Resolvers task which are located on the server.

Another library that was used was apollo-boost [Met18] which is an extension of Apollo which in turn is based on GraphQL. Apollo helps us establish the connection towards
the server. Apollo-boost is specifically made for communication with a GraphQL server and provides various tools for that.

## 5.3 Choice of Database

When considering choices for backend there were several aspects to take into account. The backend was required to have access to a database where information will be frequently updated. In the decision of what kind of database to use discussions where made whether to use a relational or non-relational database.

Apache Derby [Apa18a] is an open source relational database based on the programming language Java. It is based on Java, JDBC (The Java Database Connectivity) and SQL standards. JDBC is the industrial standard of the connection between Java and SQL databases.

MongoDB [Mon18] is a database which was used to store information in a format with attribute-value pairs and array data types. MySQL [Ora18] on the other hand is a Relational database which focuses on having a relation between different tables stored in the database which makes it easy to navigate through the different tables.

In our case, the database we chose to support the application was MySQL. One reason for using MySQL was because of the relational aspect. With a relational database different types of information are linked and structured together in a convenient way. One example of this could be having a match between two teams in an arbitrary sport. There could be many matches and each match could contain two teams. Using a relational database allows having matches reference the list of teams and it would not be necessary to store the information about the teams twice. Another reason to use MySQL was since it was known that later on there could potentially be an implementation of registering users using the application. The registered users might have their own records and information stored about them which may be connected to other areas such as schedules. Although these things could be done in other databases but differently, MySQL was deemed a good choice that would satisfy our needs.

## 5.4 Choice of Server Host

Simply having a database is not enough. A server is needed to host the database and make it accessible. Setting up a server can either be done by using a home computer or by using a cloud hosted computer in which the server can reside. In our case, setting up a home computer server that meets the requirements would be time consuming and costly.
Network security must also be taken into consideration. Therefore a cloud hosted server would be optimal in order to minimize risks and for its accessibility. When making a decision for which cloud hosted server to use, some of the options that were looked into were DigitalOcean [Dig18b], AWS [AWS18] and Firebase [Goo18b].

AWS (Amazon Web Services) is the world's largest cloud infrastructure. They offer high durability, security and flexibility to mention a few qualities. They use so-called buckets where the customer can store their data in objects consisting of metadata and object data. This service is not specifically made for developers but can be used by all kinds of different customers.

Firebase is developed by Google and is a tool to help developers build applications. It gives a lot of functionality in regards to hosting databases and gathering analytics. Firebase offers among other things a NoSQL realtime database which was taken into consideration as a candidate to use as our database although never used. They also host cloud storing, authorization, cloud messaging and more. Their service is JavaScript based in a Node.js environment.

In our case we choose DigitalOcean as the cloud hosting service because of its easy availability to develop as a team as well as high built-in security [Dig18a] and an uptime of 99.99% [Dig18b]. Uptime is the time a server is running and available. Digital Ocean is made to meet the needs of the developers. Another reason for building our server using Digital Ocean was the easy accessibility of simply having a cloud hosted computer. This meant that more of the server implementation could be handled by ourselves as well.

### 6 System Structure

As seen in Figure 1, the user will communicate with a server in order to gain the necessary updates from the database. The administrator will hold the same read privileges as the user, as well as being able to communicate with the server to update the database with new information. This could for example be setting the result of a completed game in one of the sports. The update in turn will be made available from the database to the application. The server will therefore act as an intermediate between users and the administrator and manage the different actions that the users can take.

The database will be hosted on the same unit as the server and manage all the data that will be interchangeable during the event. General information that does not change during the event will be residing directly in the application itself. Some examples of this kind of information would be rules related to different sports or the text in the menu.
buttons. A slightly more detailed overview of the system structure can be found in Figure 2.

A MySQL database was created and placed on a server on DigitalOcean cloud services. In order for the mobile application to communicate with the database and access its contents, it must go through an Application Programming Interface (API) layer on both client and server side. The client side was implemented using Apollo-Client [Met18] which is an extension of the GraphQL API.

The server side API was implemented with pure GraphQL without any extensions in combination with the libraries “express-graphql” and “graphql-yoga”. The GraphQL server manages the requests from the Apollo-Client and communicates with MySQL to access the database, as seen in Figure 3.

The method that has been used to implement push notification in this project is the utilization of Firebase. Firebase is a mobile development platform and the Firebase Cloud Messaging is the aspect which have been utilized in this project [Goo18a]. Firebase Cloud Messaging was chosen instead of Google Cloud Messaging. The reason for this being that Google Cloud Messaging became deprecated recently [Goo18c]. Firebase
7 Requirements and Evaluation Methods

Cloud Messaging is also the only service integrated with Android in regards to push notifications. Push notifications on the iOS side on the other hand rely mostly on Apple Push Notification Service to manage the pushes [App17]. However Firebase Cloud Messaging works effectively in allowing handling of push notifications to iOS as well.

7 Requirements and Evaluation Methods

There are a few requirements the system must meet. In this section we will describe these requirements and discuss how they can be tested. The main requirements we bring up in the following sections are robustness, customizable content, usability and requirements for official publication of the application.
7 Requirements and Evaluation Methods

7.1 Robustness

The server must be robust enough to not crash when there is a lot of traffic making simultaneous requests. During the event there will be up to 1500 users. This is estimated due to the fact that there usually are up to 1000 participants at the event, added with other kinds of users that are audience and the organizer. We tested this requirement with simulated connections. There are two kinds of tests that could be applied to the system. The first one is load testing, where one evaluate the performance under expected number of connections. The other is stress testing, to see what the maximum capacity of the system is. We will only perform a stress test with 1500 simultaneous connections. The reason for this is that it is difficult estimate the average number of connections at the same time, and we will still see if the system can handle the expected maximum number of connections. To execute these tests, we used Apache JMeter [Apa18b]. Apache JMeter is an open source software for load and performance testing. During the test, the number of errors and the average CPU usage of the server will be examined.

7.2 Customizable Content

For the application to be relevant each year there exists a need to update the information. The map needs to be updated to suit the new city, there might be different sports to compete in as well as new contact information and schedules.

This year there were eight sports to compete in but this is only the minimum amount of sports. Next year there might be more sports, therefore it should be easy to add and delete sport pages. The title on the sport pages as well as the rules should be easily interchangeable. The menu options needs to be easily changed as well. Since this is going to be done by non-developers it requires a good structure to simplify the process, good variable names and a guide that is easily understood. This also applies to the need of changing icons, colors and fonts if the application if used for some other event than Studentiaden.

Since the results, schedules and map markers are imported from the database, the database also needs to be easy to use for a non-developer. Results and schedules should be able to be changed quick and easy from a smartphone by the administrator for live updates. This should be done by using a good interface that only uses regular text as input and not queries.

It is also a requirement that the push notifications are working. The administrator has to be able to publish push notifications and the user has to be able to see them and to either dismiss them or accept them for the application to open.
Since we are working with a third party stakeholder, some restraints and suggestions are influencing the systems layout. Things like fonts, color schemes and logos must be easily set and to some extent also customizable like setting a theme.

### 7.3 Usability

Easy navigation in the mobile application is important for high usability. To test this we did user tests on other students who have not seen the application beforehand. They were asked to navigate to different pages and we measured how long it took for them to get there. It should not take more than 15 seconds to navigate to a page the first time a user uses the application. The limit of 15 seconds is chosen because it took us as the developers 15 seconds to read through the whole menu carefully and to navigate to a sport page.

### 7.4 Publication Guidelines

For the application to be published on Google Play and App Store it needs to meet their requirements. Neither Google Play or App Store reveals the actual requirements for publications, however they lists guidelines. The guidelines include different checkpoints in different categories. These categories include, among others, the following:

- Visual design and user interaction
- Functionality
- Compatibility, performance and stability
- Security

We will make sure the application meets the guidelines by going though the lists of guidelines for Android [And18b] and IOS [App18b], respectively.

### 8 Database Structure and Organization of Data

MySQL was used as the database management system. MySQL is an open source relational database that allows us to organize data in tables. Our database is set up with various tables that contain information about all the different elements that will appear
9 Push Notifications

in the application. There is one table that contains data about the notifications that will
be shown on the “Newsfeed” and “Important information” page. The different fields in
this table are ID, title, description and time of being posted. ID is used to associate a
post with its information, title to tell what the post is about, description to hold all main
content of the post and the timestamp to show when it was posted. The remaining tables
are structured in a similar way as seen in Figure 4. There are also relationships between
different sports teams, matches, sports associations and results.

Figure 4 Diagram representing the structure of the database tables

9 Push Notifications

Push notifications are implemented in the application to allow real time messages in
regards to important information. They are messages that appears on the screen of the
smartphone that has the sending application installed. They can consists of videos,
pictures and/or text and can play sounds, however our notifications will only display text. When the user installs the application on their smartphone they will get a question regarding if they accept that the application sends them push notification or not. If the user accepts, the device sends an ID to either Google Cloud Messaging Server (GCM) or Apple Push Notifications Service (APNS), depending on which operating system the device is running. After that the device is registered to one of these servers and will receive notifications from the application server. Whenever an administrator wants to send a notification, the administrator sends the message to GCM or APNS and that server will forward the message to all devices that are registered as notification receivers.

There exist only one situation in the application where push notifications are used in our case. This is when the organizers of the event are posting important updates in regards to event changes during Studentiaden.

10 Using the Application and Available Actions

As mentioned in section 1, there are two types of users of the applications. The first type are participants of the event, which are competitors and audience. The other type are the administrators, i.e the organizers of the event. In this section, we will describe what actions are available in the application for each of these two types.

10.1 Available Actions for Participants

The following list of actions are available for the participants:

- Viewing the “Newsfeed” page. The Newsfeed page contains:
  - Latest message from administrators. These are showed at the top of the page.
  - Matches or other sports events that are currently being played. This is depending on which kind of sports that are involved that current year. These matches and events are marked as “live”.
  - Recently terminated matches or events with score.
- Viewing rules, schedules and results for all sports that are competed in during the event.
- Viewing a map over local areas and sports halls.
• Viewing information about organizer group with contact information.

• Viewing page of older notifications from administrators.

• Viewing the “top-list” with the currently highest ranked sport association.

10.2 Available Actions for Administrators

The administrator can see everything the regular user can see. Furthermore, they can perform the following actions.

• Change the schedule.

• Add new matches and events.

• Set matches and events to “live” or “finished”. When they are not set to live, they are not showed on the Newsfeed.

• Report the winner of a match.

• Send out messages to all users of the application.

11 Design

The layout is made with the aim to be as easy as possible to use. By using a layout similar to most mobile applications, in the sense of e.g. usage of a traditional menu bar, it becomes more intuitive and easy to learn. The title for the page is in the top alongside a menu button that shows the navigation menu when pressed. An example of this can be seen in Figure 5. Since 49 percent of smartphone users are holding the phone with one hand [Ste13] and only about 10 percent of the population is left handed [Har77] the menu button is located in the upper right corner which make it easier to press when the user hold the phone with the right hand. The menu is ordered after what will be the most visited pages in the top and the least used pages in the bottom. The “results”, “rule” and “schedule” buttons in the menu are displayed in the form of a drop-down that slides down to reveal the buttons. Using this layout might be making it harder to navigate through the menu [Jak07] slightly since the shape of the menu changes when pressing a button. However this is compensated by gaining a better overview of the choices and navigation instead. Without the drop down, the menu would be much longer and therefore more difficult to navigate through.
On the sport pages there are three buttons in a row right underneath the title that shows result, schedule and rules for the chosen sport. These three buttons are in the same color as the title background color to make the header appear as a single unit. All the pages are scrollable and the sport pages are horizontally scrollable to change pages between results, schedules and rules. The three sports buttons make it possible to access the sport pages in two ways. Either the user can choose them from the menu or they can choose a sport from the menu and then choose whether to show results, schedule or rules. This is making it easier to navigate through the application. The map page has a start position at the location of the event.

All buttons have consistent design. Color and font chosen is made according to Swedish University Sports Federation’s guidelines for design. They are all bigger than the 44 pixels that are the recommended minimum size for buttons [App18d] and they give feedback to the user when they have been pressed.

The colors (blue and white) have a contrast ratio of 7.49:1. This is higher than the minimum recommended contrast ratio of 4.5:1 [Web18a], which make it easier for colorblind people to read the text.

![Newsfeed Screen](image)

**Figure 5** Newsfeed screen
In this section, the result of our evaluations are represented. First, we discuss the results from the robustness test. Second, we will go in to whether or not the content is customisable. This is followed by a presentation of the results from the usability test. Finally, we discuss the publication requirements and how well the application meet them.
12 Evaluation Results

12.1 Robustness

The robustness was tested using JMeter [Apa18b]. A stress test was performed, while a load test was not. The reason for this is that we wanted to test the limits of the system, in the event that all expected users tried to access the server at the same time. The following parameters were initially set:

- Number of Threads (users): 1500
  - This is the number of users simulated
- Ramp-up period (in seconds): 0
  - This is the amount of time for JMeter to get all threads started
- Loop count: 1
  - How many times the test will be run.

In JMeter, different kind of samplers can be added. In this case we added a HTTP request sampler. To run the test, one also need to specify what kind of HTTP request that will be sent. We sent in a simple GET request that accesses the server. The test was then performed 20 times in a row. The test results can be found in figure 8.

In none of the tests, any error appeared. The average CPU usage was a little over 25%.

12.2 Customizable Content

There is no easy way to add or delete sport pages due to the implementation difficulty. Instead we will write a guide for the administrators that is easy to follow.

For the live updates, results and schedules, the administrator fills in a form that only consists of regular text that aims to be easy to understand by a non-developer. Unfortunately we did not run any test to see if they were understandable by non-developers.

In regards of changing the information that are not fetched from the database we have implemented a special file for all texts and icons so they are easy to change.

Push notifications has been implemented and works well in the regard that information is sent as a push to the user when information in the database is updated. The information that is updated in this case and sends push notifications is important information from the organizers.
### 12.3 Usability and Layout

The navigation test was tested on 9 people who had never seen the application before. They were asked to navigate to a page two or three clicks away from the home screen (e.g. first click on menu button and second click on some page on the menu). The average time it took for them to get to the desired page was 10 seconds, with the fastest time on 8 seconds and the slowest 13 seconds. Testers that were asked to navigate to a page that required three clicks took longer time than the ones that were asked to navigate to a page that only required two clicks. We also saw that every button was easy to press. All test takers understood that the buttons actually were buttons despite them having the same color as the background.

![Figure 8 Compilation of JMeter test results over 20 run tests](image)
12.4 Publication Guidelines

The application was not published during the time of this project due to long waiting time for publication on Google Play and App Store. Therefore we do not know if the application met all the requirements for publication or not. However, we have evaluated our application to the guideline checklist. The application satisfies almost all the criteria in the guidelines. It satisfies that normal functions, like the back button or the button to turn off the screen is not redefined to trigger some other function. The design are not misleading or uses icons that normally uses for some other function, like arrows, menu icons and such.

The application only requests for permissions that it actually needs, like the permission to display notifications and not the permission to use the camera. No sensitive information is stored and all information that the application is downloading is put into local files that only this application can access.

One guideline that the application does not satisfy was the lack of a working layout when the user turns the phone to landscape mode. The reason for that was that the landscape mode feature was disabled which was acceptable. Another guideline that cannot be met is that push notifications are only being used to convey information related to the applications content. This is out of our hands to control since we are not the ones in charge of handling the push notifications. The push notifications are managed by the organizers who post the information. This information could have been made irrelevant or inappropriate.

13 Results and Discussion

This section is divided into two parts. The first part will discuss how well the application met our requirements and bring up some of the reasons we think it did or did not do so. It will also bring up the discussion of why the results ended up as they did, and what might have been done to improve them. We will in the following part consider the results to see if the application was useful for the distribution of information in the case of a sports event.

13.1 Satisfying the Requirements

One requirement we had on the system was that it needed to be robust enough to handle up to 1,500 active users at the same time. The result of our stress test showed that the
server could indeed handle 1 500 simultaneous connection. This is since there was no errors during the 20 times the test was run. The average CPU usage altered between 13.5 % and 28.3 during the tests, with an average of 25.05 %. This means that the robustness seems to be satisfying enough for the application to be used during Studentiaden.

That is with the current amount of expected participants, with the estimated number of audience. The number of users could however be more or less than we estimated for our tests. However we believe that 1 500 connections are an overestimation. There is a high probability that the actual number of users is less than the number of users we have tested for. For a more accurate testing, there is also some things to consider. The test was run in GUI (graphical user interface) mode. It is however more efficient to run the test in non-GUI mode, in the terminal. This is since GUI consumes resources, which could impacts the test result. The test also uses a simple GET request. It would be more realistic to send in a real query, or make a separate test for accessing the database.

Furthermore, we wanted the application to be customizable. This means that the application could be used from year to year, even though the city and the sports changes. In the current version, the static content in the application is adjustable through changing a file. This is not optimal, since the application will need to be re-uploaded to App Store and Google Play each year. It is also not optimal for a non-developer to change the content, since it requires some knowledge about the file structure and where to edit. This might be a problem even though we attach instructions and documentation to our stakeholder. A better approach would be to implement an interface through a web page to allow full maintenance of the applications icons, text, and other features. This ended up being a limitation or a regard for future work.

The results of the usability test showed that it was easy to navigate in the application. We received some comments on the layout. One tester suggested another way to navigate through the application. However we believe that our way to navigate is more intuitive since it uses a menu for navigation, which is familiar to most users. We also took in to the consideration that no other test person had any comments on the navigation. The number of tested people should be enough to get insights on what we need to improve since according to NNgroup only 5 test persons is enough for this kind of study [Jak12]. However we did not test the usability of the administrator version of the application. The reason for this is that the version was incomplete by the end of the project. Therefore it was not possible to evaluate the full scope of the administrator application. The navigation to the different sites will however be the same, since they are built upon the same shell. The administrator will have some additional available actions, listed in section 10.2. Because of this, it would be interesting to test how intuitive it would be to perform these actions.

The guidelines for publication on Google Play and App store were met to a satisfying
level. There were some of them that were not met, like horizontal mode or control over the push notifications. However we have no reason to believe that the push notifications will be misused by the administrator.

In general, the evaluation results shows that there is room for improvements and further testing. But after a few alterations, we believe that the application could be published and used during upcoming events.

13.2 Using the Application as a Source of Information

To determine for sure if the application would be a good source of information during an ongoing event, it would have to be tested in a real environment. We believe that it will satisfy the need of information updates. We are able to load and modify content in the database, such as sports results. We were also able to send out push notifications from the Firebase messaging system. These actions was some of the main purposes of the application, and they are working satisfactory. The application could however be seen as a complement to more traditional information distribution methods, such as posters, more than a replacement. That is since not all people are using a smartphone, or using the operating systems that we have developed for. The usage is also dependent on whether or not the organizers inform enough about the existence of the application before the event. Because of these various reasons, results and general information could with advantage be published on a website as well. The application could then be further elaborated to be connected to such website, so that the same information will not have to be published twice. This is further discussed in section 15. The main advantage of using the application during the event is that both results and information from organizer are available in the same place. Since we implemented push notifications, all users of the application will get a notification each time a new message is being sent. This leads to a more efficient information distribution.

14 Conclusions

Although the application was never posted on the application markets, the results from the local development version of the application were still enough to give a good hint at answering some of our questions of using a mobile application as a means for distributing information. The application can indeed be used to convey information both in a static way similar to how analogue posters do, but also more dynamically with the use of push notifications. Assuming most people that participates in the event has downloaded the application, any new information that may be important to them will be
available fairly shortly after it has been sent by an administrator. In other words, a large amount of people will be notified in a short time. Even if only half of the participants download the application, chances are that the information will also be spread by word of mouth as in with posters. The application should therefore be a viable option for spreading information to a large group of people.

But how effective is it? The server passed the test of 1 500 simultaneous connections which suggest that a large group of users could use the application simultaneously. This could very well be useful in events like Studentiaden where the information is not in particularly high demand until the event is underway. In comparison to the other ways to communicate during Studentiaden as mentioned previously, we believe that using an application like the one that was developed may be one of the most effective ways to distribute information for events like Studentiaden, provided the users are willing to download an application.

15 Future Work

There are several ways the mobile application can be further developed. In this sections, these possibilities are described.

As discussed in section 13.2, it is probable that there still exists a need of a website, even if the application is used as intended. One future development is to connect the website and the application. The purpose of doing so would be to only having to update the information in one place, and it would then show up both on the website and in the application. To facilitate for the participants to get relevant information, the ability to create personal accounts could be implemented. This could also be created from the information given at registration. The users would then get the schedule for their teams matches automatically.

Another way to improve the application is by allowing the user to mark events to make them show in a personalized schedule. It could also be possible to filter the schedules by association. This would minimize the time required to find information about a desired game during the event. It is not optimal to have to scan through every game to find the desired one.

At the end of this project, the application is only available in English. Before publication, the default language will be changed to Swedish, since our main target audience are Swedish speakers. This change would be easy and fast to implement since the text strings that is showed e.g. in the menu exists in a limited amount of places in the code. An future improvement is to allow the user to switch between languages, like English...
and Swedish or other desired languages.

It could be useful to integrate the application with the public transport timetables to make it easier for the participants to find their way around the city.

Another way to improve the application to help the users is to implement a page for frequent asked questions that could be updated during the event. This could help participants to get answers to their questions faster and ease the workload of the organizers.

When a game is being played it is displayed in the News feed page as “live” but the results are not being updated live. The results are reported when the game has finished. Here an improvement could be made, by allowing the live results to be updated right away.
References


References


References


