Candidate number: _____

BACHELOR'S THESIS

ProAct – digitizing paper-based tools for municipal health care

Håkon Hole Lønning and Juan Carlos Arias Holguín

Information Technology

Department of Computing, Mathematics and Physics

Supervisor: Carsten Gunnar Helgesen

Submission Date: 03.06.2019

I confirm that the work is self-prepared and that references/source references to all sources used in the work are provided, cf. Regulation relating to academic studies and examinations at the Western Norway University of Applied Sciences (HVL), § 10.

Høgskulen på Vestlandet

Faculty of Engineering and Science
Department of Computing, Mathematics and Physics

TITTELSIDE FOR HOVEDPROSJEKT

Rapportens tittel: ProAct - digitalisering av papirbaserte verktøy i det kommunale helsevesenet	*Dato:* *03.06.2019*
Forfatter(e): Håkon Hole Lønning, Juan Carlos Arias Holguín	*Antall sider u/vedlegg:53* *Antall sider vedlegg: 2*
Studieretning: Informasjonsteknologi	*Antall disketter/CD-er:* 0
Kontaktperson ved studieretning: *Carsten Helgesen*	*Gradering:* Ingen
Merknader:	

Oppdragsgiver: Bergen Kommune	*Oppdragsgivers referanse:*
Oppdragsgivers kontaktperson: Kurt Arild Krokmyrdal	*Telefon:* 930 02 455

Sammendrag:

Stikkord:

Høgskulen på Vestlandet, Fakultet for ingeniør- og natuvitskap
Postadresse: Postboks 7030, 5020 BERGEN Besøksadresse: Inndalsveien 28, Bergen
Tlf. 55 58 75 00 Fax 55 58 77 90 E-post: post@hvl.no Hjemmeside: http://www.hvl.no

PREFACE

This report is about the bachelor thesis in Software Development by Håkon Hole Lønning, student at the Western Norway University of Applied Sciences; and Juan Carlos Arias Holguín, exchange student from the University of Extremadura (Spain). We have been working on this project during the last three months, full time developing an application for the Bergen Municipality ER.

This project has been a learning experience for both of us, since we have experienced a work environment very close to a professional one. We have applied many of the knowledge acquired during the bachelor's degree, in areas such as planning, management and development.

We would like to thank Bergen Municipality ER for the opportunity to work with them on such a big project; and its employees, for their continued encouragement and constructive feedback.

Finally, we would also like to thank our coordinator, Carsten Gunnar Helgesen, for his help during all the development of the application and his suggestions for the structure of this document.

Håkon and Juan Carlos.

TABLE OF CONTENTS

TERM LIST

Activity

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI (Google 2019, b)

Intent

An intent is an abstract description of an operation to be performed. An Intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is basically a passive data structure holding an abstract description of an action to be performed. (Google 2019, e)

Bundle

A mapping from String keys to various parsable values. (Google 2019, g)

Layout

A layout defines the structure for a user interface in an app, such as in an activity. All elements in the layout are built using a hierarchy of View and ViewGroup objects. A View usually draws something the user can see and interact with. Whereas a ViewGroup is an invisible container that defines the layout structure for View and other ViewGroup objects, as shown in figure 1. (Google 2019, c)

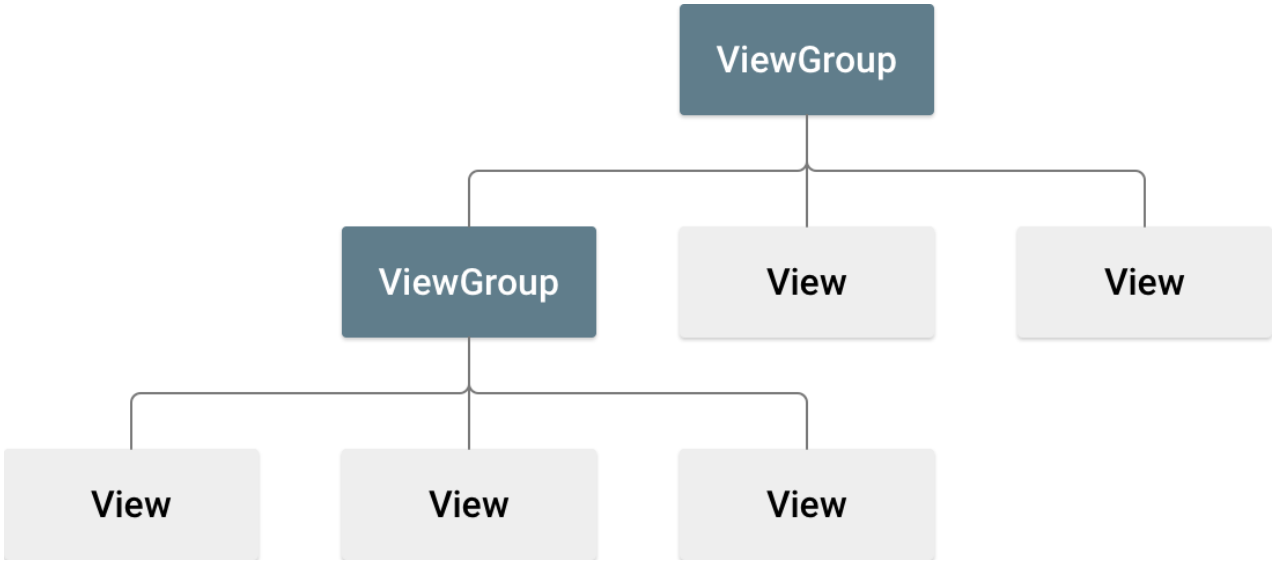


Figure 1- ViewGroup objects

Constraint Layout

Constraint Layouts allows us to create large and complex layouts with a flat view hierarchy (no nested view groups). It's like RelativeLayout in that all views are laid out according to relationships between sibling views and the parent layout, but it's more flexible than RelativeLayout and easier to use with Android Studio's Layout Editor. (Google 2019, d)

Material Design

"Material Design is a visual language that synthesizes the classic principles of good design with the innovation of technology and science" (Google 2019, a)

String resource

A string resource provides text strings for your application with optional text styling and formatting. (Google 2019, f)

Toast

A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout. (Google 2019, k)

RecyclerView

A flexible view for providing a limited window into a large data set. (Google 2019, l)

Persistence

In computer science, persistence refers to the characteristic of state that outlives the process that created it. This is achieved in practice by storing the state as data in computer data storage. Programs must transfer data to and from storage devices and have to provide mappings from the native programming-language data structures to the storage device data structures. (Balzer, 2005)

Entity

Can also be a database table, with or without relations. Each entity must have at least 1 field annotated with PrimaryKey. (Google, 2019, m)

Foreign key

Foreign keys allows you to specify constraints across Entities such that SQLite will ensure that the relationship is valid when you modify the database. (Google, 2019, n)

Primary key

Each entity must have a primary key, which states how to look up tuples and/or attributes in a database table. Without a primary key, the database structure is undefined and persistence cannot be attained. (Google, 2019, o)

Repository

Repository modules handle data operations. They provide a clean API so that the rest of the app can retrieve this data easily. They know where to get the data from and what API calls to make when data is updated. You can consider repositories to be mediators between different data sources, such as persistent models, web services, and caches. (Google, 2019, j)

DAO - Data access object

A Database Access Object is an abstract interface that states how to perform Create, Read, Update, Delete (CRUD) operations for an entity in a database. You use the DAO, its methods and properties to manipulate an open database.

Database

A relational database (RDB) is a collective set of multiple data sets organized by tables, records and columns. RDBs establish a well-defined relationship between database tables. Tables communicate and share information, which facilitates data searchability, organization and reporting.

RDBs use Structured Query Language (SQL, /ˈsiːkwəl/ which is a standard user application that provides an easy programming interface for database interaction. (Techopedia, 2019)

1. INTRODUCTION

1.1 Context and motivation

Triage Early Warning Score (Wallis et al. 2006):

> *Physiological assessment was chosen as a major component of the system as it is a core element of triage. The Medical Early Warning Score (MEWS) utilises systolic blood pressure, heart rate, temperature, respiratory rate and AVPU (a measure of level of consciousness, viz. Alert/Verbal/Pain/Unresponsive) as parameters. MEWS has been used to successfully identify physiological deterioration of medical inpatients, where MEWS scores of 5 or more were associated with increased risk of death, ICU and high dependency unit admission. The MEWS score identifies patients who need medical intervention. However, the MEWS has limitations with regard to triage in that it is medically biased [...] The addition of both a mobility parameter and a trauma factor increases the severity score for trauma patients, as well as for medical patients who are physiologically normal but have time-critical conditions [...] These parameters have therefore been added to the MEWS score [...] and the resulting system has been renamed the Triage Early Warning Score (TEWS)"*

ADULT TRIAGE EARLY WARNING SCORE (older than 18 years)							
	3	2	1	0	1	2	3
Mobility level				Walking	With help	Stretcher/ Immobile	
Respiratory Rate		≤ 8		9-20		21-30	> 30
Heart Rate		≤ 40	41-50	51-100	101-110	111-130	> 130
Systolic Blood Pressure	≤ 70	71-80	81-100	101-139	140-199	> 199	
Temperature		Feels cold OR ≤ 35°C		35°-38.4°C		Feels hot OR > 38.4°C	
Level of consciousness				Alert	Reacts to Voice	Reacts to Pain	Unresponsive
Trauma				No	Yes		

Figure 2 - The TEWS scoring system

and National Early Warning Score 2 (Royal College of Physicians, 2017):

> *"NEWS2 is the latest version of the National Early Warning Score (NEWS), first produced in 2012 and updated in December 2017, which advocates a system to standardise the assessment and response to acute illness.*
>
> *The NEWS is based on a simple aggregate scoring system in which a score is allocated to physiological measurements, already recorded in routine practice, when patients present to, or are being monitored in hospital. Six simple physiological parameters form the basis of the scoring system: respiration rate, oxygen saturation, systolic blood pressure, pulse rate, level of consciousness and temperature.*
>
> *A score is allocated to each parameter as they are measured, with the magnitude of the score reflecting how extremely the parameter varies from the norm."*

Chart 1: The NEWS scoring system

Physiological parameter	Score 3	2	1	0	1	2	3
Respiration rate (per minute)	≤8		9–11	12–20		21–24	≥25
SpO_2 Scale 1 (%)	≤91	92–93	94–95	≥96			
SpO_2 Scale 2 (%)	≤83	84–85	86–87	88–92 ≥93 on air	93–94 on oxygen	95–96 on oxygen	≥97 on oxygen
Air or oxygen?		Oxygen		Air			
Systolic blood pressure (mmHg)	≤90	91–100	101–110	111–219			≥220
Pulse (per minute)	≤40		41–50	51–90	91–110	111–130	≥131
Consciousness				Alert			CVPU
Temperature (°C)	≤35.0		35.1–36.0	36.1–38.0	38.1–39.0	≥39.1	

Figure 3 - The NEWS2 scoring system

are tools to assess a patient's general condition by measuring vital data and using it to calculate a score that tells the health care personnel something about the patient's need for

care and monitoring. It may also tell health care personnel about what the patient's risk for circulatory failure is. Every patient at the municipal hospital will be assessed at least once per day using this sheet.

1.2 Goal

The goal of the project is to replace Bergen Municipality's current paper-based system of registering, calculating and storing ***NEWS2*** and ***TEWS*** scores for their patients in the home care service and at the municipal hospital with an Android application.
When using the Android application, a user will input vital data from a patient and get a TEWS or NEWS2 score back. This application will be able to bring back up historical data for a patient in order to see changes in the patient's general condition over time. For this project to be worthwhile the user experience needs to be intuitive and user friendly, such that users will prefer to use this application as opposed to manual calculation.

1.3 Limitations

The main limitation we have with this project is using the production environment. The municipal network structure and authentication service is proprietary and needs a lot of work with integration for it to be operational. Our current timeframe is not enough for such a task. Therefore, we have focused on delivering a product that can be integrated with the municipal network structure in posterity by other parties.

1.4 Resources

We will need many kinds of resources for our work. Here follows a specification of what we have ascertained that we will need.

1.4.1 Information need

With the first meetings done with our client, we have established a framework for what the finished product should be. The operating system we will work with is Android for cell phones and tablets. This means we already know something about what to expect when choosing development tools. The project group already has some experience with using Android Studio for developing Android applications using Java. Thus, we have a good starting point for working with these tools.

We will need to know the client's expectations and requirements for the finished product. We have already established a foundational framework through meetings with the municipal operations department and with a clinical nurse specialist at the municipal hospital. To ensure we are not deviating from their expectations and requirements in the future, we will arrange meetings with a focus group of 5 people from the hospital every 3 weeks until the building phase is completed.

1.4.2 Testing

We will be using the two major tests available for application development, which is technical and functional testing.

For technical testing in Java, one frequently used framework is JUnit. It has many built-in functions that test code block output and behaviour. One thing technical testing will not

catch is if the developer has produced logical errors that produce exactly the result intended, but it will produce a wrong result in conjunction with the other parts of the application. This is where functional testing may be able to catch what the technical tests could not. Functional testing basically means that a user will use the application and see that it does exactly what was intended and nothing else.

Every iteration of the project will end with thorough testing to ensure further work does not build upon wrongful code.

As the last step of each iteration we will present the work to a focus group for final assessment and feedback. This feedback will be a vital component for further work in the next iterative phase.

1.4.3 Development

We will need to plan what technologies that will be needed. As this will be a mobile application for Android phones and tablets, the most reasonable choice of programming language will be Java, as it is a language known by both members of this project group and it is a native language of Android.

We will need to decide what frameworks that will be needed for our application functionality and design as well. This is a decision that we will have to decide further into the development process.

One major part of development is planning. There are many ways to plan software development. We decided upon an iterative development plan that will need to be detailed further.

1.4.4 Human resources

As we develop a mobile application, we will need professional help with administration, writing, software development and any other areas touched by this project. For this we have the possibility of contacting internal lecturers and professors at Western Norway University of Applied Sciences for technical questions. We have the possibility of contacting the client for questions regarding scope details or product specifications. The focus group at the municipal hospital will be contacted for feedback at the end of every development iteration.

Our thesis coordinator is an important guide to questions we had not thought of and pointers to where our priorities should be.

1.5 Organization of the report

The report starts with a list of terms used throughout the report, providing context and definitions to these terms.

Chapter 1 defines the context, motivation, goals and limitations of the project. It also explains why Bergen Municipality wants the project

Chapter 2 gives a description of previous work the application builds on. Initial requirements are structure based on how Bergen Municipality envision the application.

Chapter 3 gives a structured explanation of the project. Different approaches are discussed, and one is selected. A plan is formed through several use-cases, detailing the method of development, the tools to be used and the evaluation methods.

Chapter 4 explains the development process. The architecture (both internal and external) is discussed and elaborated. It also explains some thoughts about user experience design and choices made. The most important parts of the application are expressed in detail and a walkthrough of how the application works is given.

Chapter 5 is about the evaluation methods used and the results that have been extracted.

Chapter 6 presents the overall results of the project.

Chapter 7 gives a discussion about choices made during the development.

Finally, conclusions are drawn in Chapter 8

2. PROJECT DESCRIPTION

2.1 Practical background

Bergen Municipality delivers the frame to work with. On the very first meeting, held in early February, the employers gave us the tools to work with and explained us how to use them, but there are currently no proper technology solutions based on them.

2.1.1 Project owner

The students own the project but has made it open source, such that other parties willing to continue using their work are able.

2.1.2 Previous work

We got access to some previous work done by a student group; it was about a similar project to this. The work was mostly pertaining to business models and plans. There were not any design solutions or practical information present. There was a risk analysis and some time frame planning. It must be noted that the previous work was revolving around technology leadership and not programming.

2.1.3 Initial requirements specification

The project needs to solve the following:

- Should be possible to choose between both NEWS2 and TEWS
- Should be possible to gather all values for each test
- Should be possible to get the score automatically from those values
- Should be possible to compare the last score with previous ones
- Should be possible to gather all that information into Bergen Municipality's database system

2.1.4 Initial solution idea

The first idea that came to our mind was a mobile application which can be used to solve those tasks quicker and easier than doing them on paper. This solution has an extremely simple user interface which will provide all the functionality required. So, we asked our employer for a meeting with the focus group to show them a prototype of the application. This first meeting with the focus group was held in the first stage of development.

The project group displayed some sketches of possible user interfaces, designed according to the feedback from the first meeting with the employer. The coordinator pressed the importance of not thinking too big. The consensus between all parties was that the user experience should be the focus such that users would rather use this new tool instead of calculating everything manually.

This prototype was an android application that allows the user to choose between the two tests, collect values and show the final score.

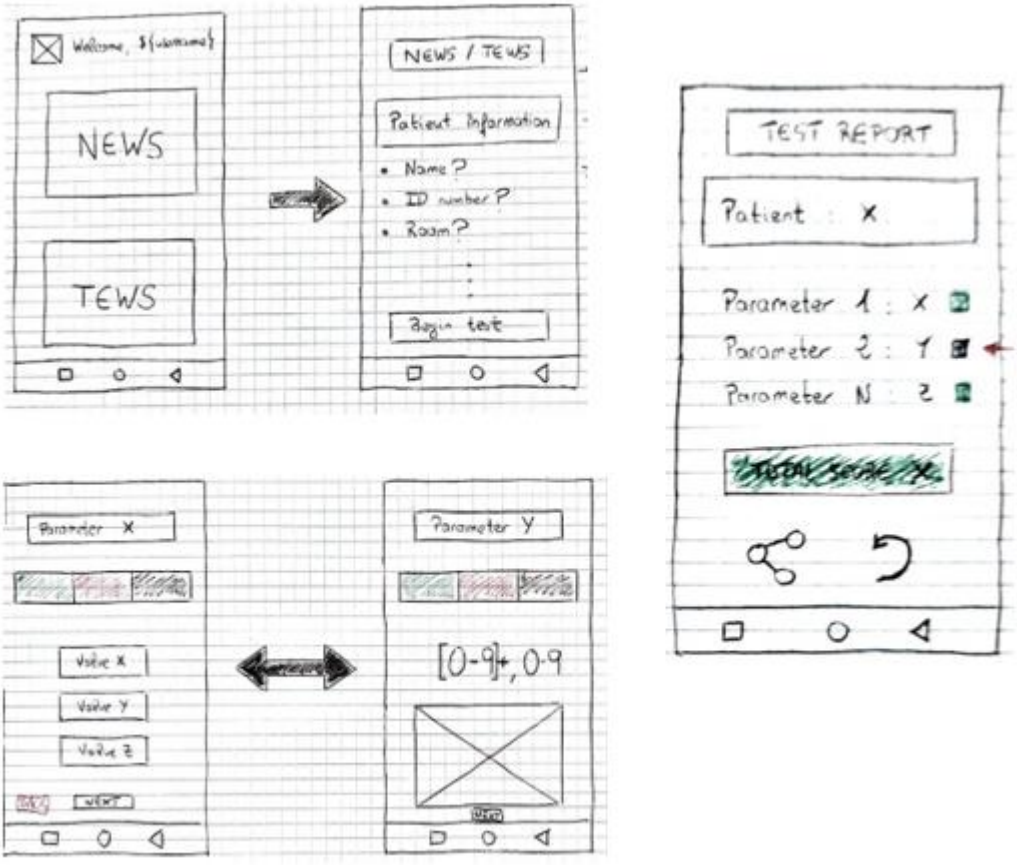


Figure 4 - Sketch of the user interface prototype

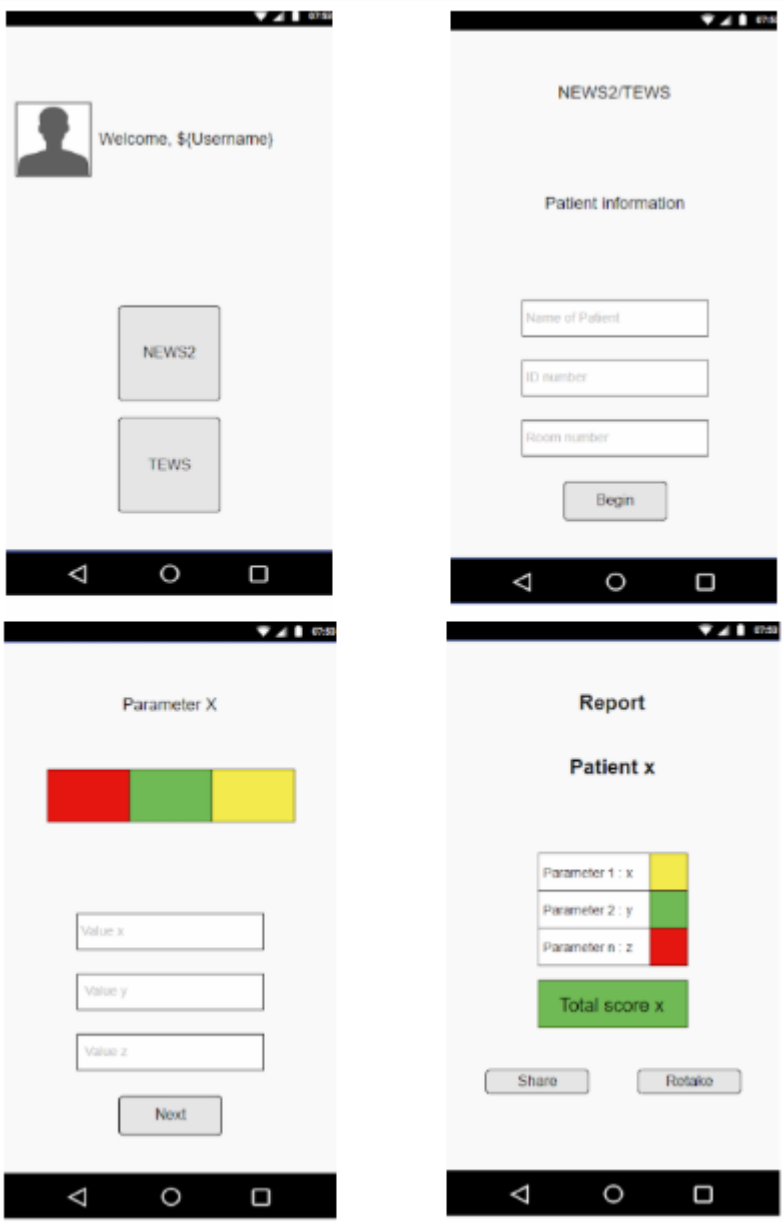


Figure 5 - Digitized sketch of the user interface

3. PROJECT DESIGN

3.1 Possible approaches

Several approaches can be considered when designing a mobile application. These approaches all offer different benefits, but also drawbacks compared to each other. Following is an examination of these approaches and a subsequent discussion.

3.1.1 Alternative approach 1 - Web

A web application is essentially a website that can be tailor made for any device dynamically. They can be made using many different languages and frameworks, like JavaScript/React/Node.js, C#/.Net, Java EE, Python, PHP, etc. Web applications are accessed through a web browser like any other website. However, they can be designed to provide the user with the navigation capabilities and options of a mobile touch-based device.

Web applications are a good solution because they can be developed quickly and can reuse code internally, such that you can easily change the appearance of views and there will be less boiler plate code. But it is not a good practice to develop a webpage to be used on a mobile device, because they cannot use some of the powerful functionality of a mobile device, such as portability and data inherent to the device itself.

3.1.2 Alternative approach 2 - Native

Native applications are written for the specific platform with direct access to that platforms SDK and thus offers direct access to the device platform. This approach gives access to the newest functionality on each platform. However, it also requires that an application is made for each individual platform. This requires the project team to maintain a lot more knowledge and skill in various programming languages if it wishes to create multi-platform applications.

3.1.3 Alternative approach 3 – Cross platform

A cross platform application is designed to be translated into platform specific code, allowing the project to have a single code base for multiple platforms, increasing code-reusability.

A cross platform approach means that the developer must be able to fulfil user experience requirements for every platform intended. Failing on meeting these standards will lead the user to a frustrating experience.

Basically, when you think about a cross platform approach you need to be aware of many user experiences and functions as many platforms you want your application to be working

at. This would require an extremely large amount of testing and tinkering to make the applications work appropriately and as specified.

3.1.4 Discussion of alternative approaches.

For this thesis we face mainly two problems: lack of time and lack of knowledge. We have very little time to develop an application by using technology fairly new to us, so we must work while learning most of the time. That is the reason why we will be focusing on delivering a fully functional product, even if it entails renouncing certain requirements. We would prefer working with something the team already has knowledge of, and knows how to use, to be able to deliver a product.

The municipality wished for more functionality than we were comfortable with including, such as integration with their own database system, which would take time the project could not handle.

For us, using a native only approach to Android's development will save a lot of time. This is mainly because both of us are familiar with the native Android development language, Java.

3.2 Specification

We are aiming our work to reach the project goal, and this is best done by cutting risks and building on knowledge we already possess. Even though the finished product will only run on the Android platform, we believe that a native approach will maximize our chances of success.

3.3 Selection of tools and programming languages

We will be using Android Studio IDE.

> "*Android Studio provides the fastest tools for building apps on every type of Android device. Android Studio contains a lot of premade modules and allows us to generate code automatically for some repetitive tasks. There are a few alternatives but none that offers as many features as Android Studio*" (Google 2019, i).

For the persistence of our application, we will be using Room Persistence Library.

> "*The Room persistence library provides an abstraction layer over SQLite to allow for more robust database access while harnessing the full power of SQLite".* (Google 2019, h)

3.4 Project development method

3.4.1 Development method

For this project we have chosen to develop using an agile approach to software development through an iterative model. It will allow flexibility approaching the project and being able to handle problems as soon as they show up.

We will work next to each other. By working together, we can help each other easily. We decided that the most efficient thing would be for each group member to specialize in a specific development branch, based on our previous experience as programmers.

- Juan Carlos Arias Holguín will work mostly front-end, application design and user experience focus
- Håkon Hole Lønning will work mostly back-end, database integration and architecture design.

Sectioning the development into frontend and backend creates a clear definition of responsibility. This makes it easier for the team members to create intermediary objectives, which is the backbone of iterative development. This choice of development strategy suits a small team of only two members better than other strategies, such as the lean development or the waterfall method.

We adopted an agile form of development called Kanban. Where we had a Kanban board in the form of a Google Docs document where we could insert "TODO" items and mark them as resolved or in progress in real time. This was very useful for delegation of workload and for keeping up to date on how the other part of the team was progressing.

As this project consisted of a team of two, keeping á jour with each other's progress was important in order to avoid mismanagement of time, as well as conflicts in the source code.

We used a team foundation called bitbucket for collaboration and commits. Using a team management tool for source code is very useful when cooperating with others. It is easy to create a new branch from the code and work on it without getting merge conflicts when others are writing on the same code.

3.4.2 Project Plan

In order to reach the deadlines, we must get an overview of what needs to be done and in which order. This started with conversations with Carsten, who gave us some advice to get started with. This, plus the deadlines from HVL, gave us a general timeline.

For this pre-project report we will split the project in four phases, just to make it clearer what the current focus is:

1. ***Pre-planning:*** choosing a project to work on, meetings with both coordinator and employer
2. ***Pre-project and prototype:*** exploration of different solutions and showing them to the employer
3. ***Development:*** build the prototype, making iterations based on continuous feedback
4. ***Evaluations:*** test our prototypes and get feedback from the focus group

5. Report and handover

3.4.3 Risk management

In order to properly handle the various risks, they need to be identified and classified. This work begun in the pre-project stage but also continues into the later project phases. Managing a risk is a matter of minimising the risk impact on the project in such a matter that the risk can be neutralised as soon as possible. Classifying a risk makes it easier for the project to avoid them, and we ensure that risks will be handled by priority. By working with iterations, we can easily tell if a risk is handled and mitigate the risks as soon as they appear.

3.5 Evaluation method

We concluded that nurses and doctors would be the best users to test the application, since they knew the procedures and held the experience needed to give us feedback on the project's success.

The lack of time is a major drawback as we will not be able to get as much feedback as we would want. To mitigate this, we will attempt to schedule as many meetings as possible to use that feedback towards the next iteration. This creates a circle of development and evaluation helping to ensure a good project progress.

4. DETAILED DESIGN

This chapter will examine in-depth every system component. Since we rely on modularity as an essential feature of software development, every module examined here is relevant for the functionality of the application.

4.1 Use-cases

Earlier in this report we discussed which topics should be the focus for this project. These topics will be useful for getting a better understanding of the problems that the application is supposed to solve.

To understand the purpose of all the NEWS2 and TEWS functionality and requirements, we needed to arrange a few meetings with our employer and the focus-group of Bergen Municipality. From these meetings we made some of the use-cases that we concluded would be most crucial to solve. It is important to mention that, even though we found several potential use-cases, we are only going to discuss those we have concluded will be most crucial. This is because our project does not meet the time requirements needed for a wide scope. Since this is a "three effective months" project, and only one month could be

used for effective programming and developing of the application, we had to limit our scope to be able to make something realistic and manageable.

From the early project description, the feedback from the employer and focus group, and the work done on previous versions of the report, we concluded that our focus should be the user experience.

We defined the following use-cases:

- Choose what test to do
- Get the score for a patient
- Get a summary of every test done.

The core of these use-cases is that functionality is the driver of the application. The main purpose of the application is to automate the NEWS2 and TEWS paper-based systems, such that this is the main use-case. As those systems are quite similar but still different, the user must be able to choose which one to use. This is discussed previously in the report. Finally, the point of using those systems is to be able to see how a patient is progressing, therefore it is a requirement that the users must be able to bring up past results and compare them as well.

Use-case 1: Choose between NEWS2 and TEWS

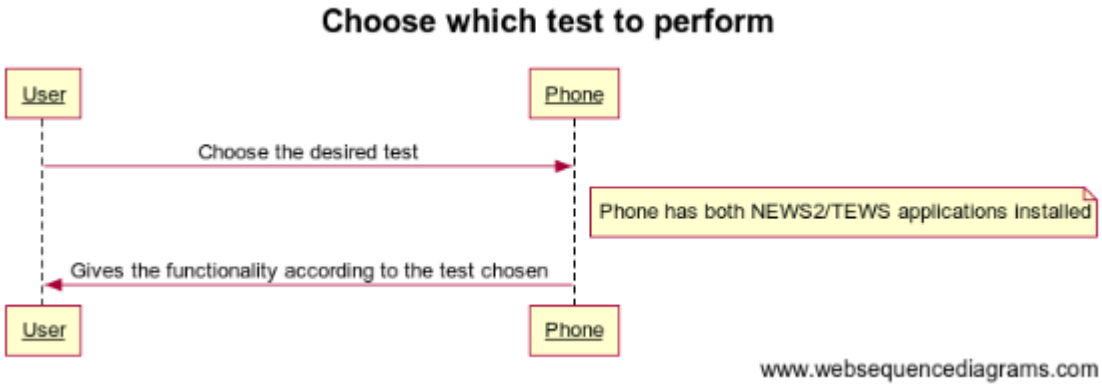


Figure 6 - Use-case 1: Choose between tests

This use-case seems obvious, but it is essential because users must be able to easily choose what system they need at every moment. As discussed previously in this report, the first prototype of the application was designed to have both tests inside the same application and let the user choose what test to perform on the menu screen. However, during one of the feedback meetings, the focus group informed us that they found this design to be quite inefficient, as it could lead to confusion, therefore they asked us to split the project into two applications. This turned out to be a change for the better for the application's design, since it proved useful in terms of reusability. As both tests are quite similar, once finished with the first one we merely had to copy and paste most of the code from the second one. When it comes to maintenance it is far more efficient to have two applications. If you want to fix or improve one you do not have to worry about breaking the other one.

Use-case 2: Get the score for a patient

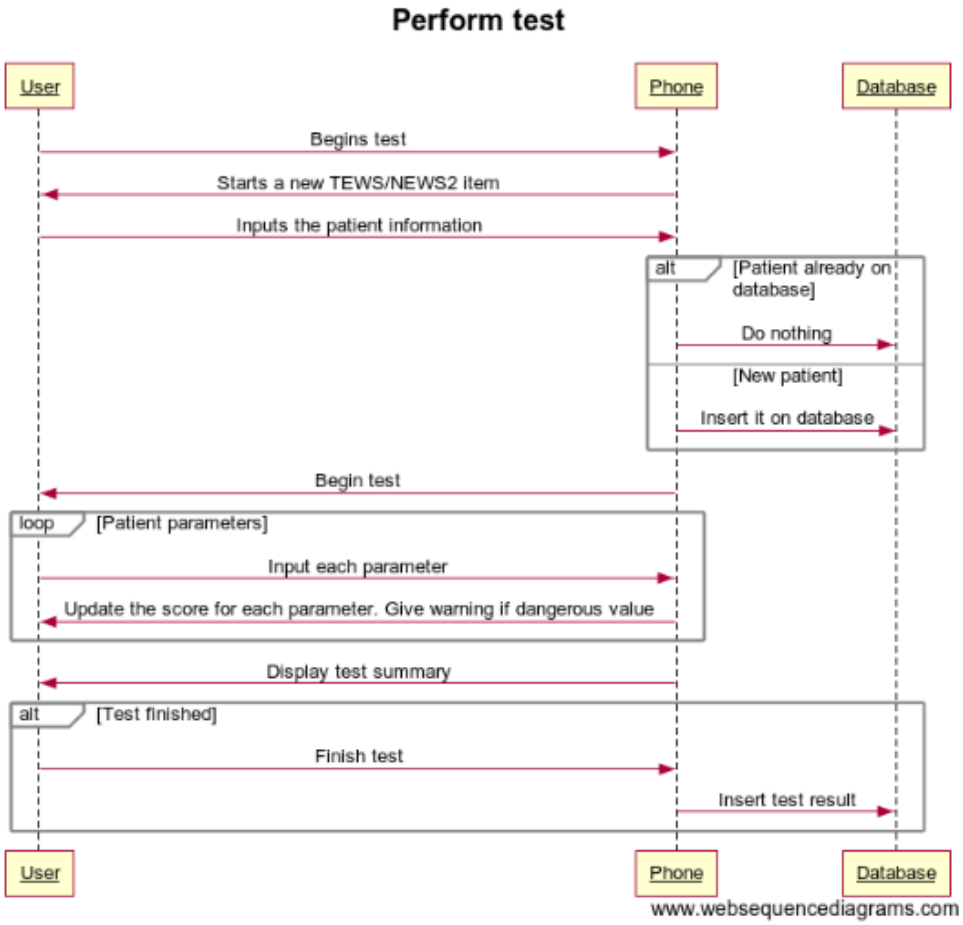
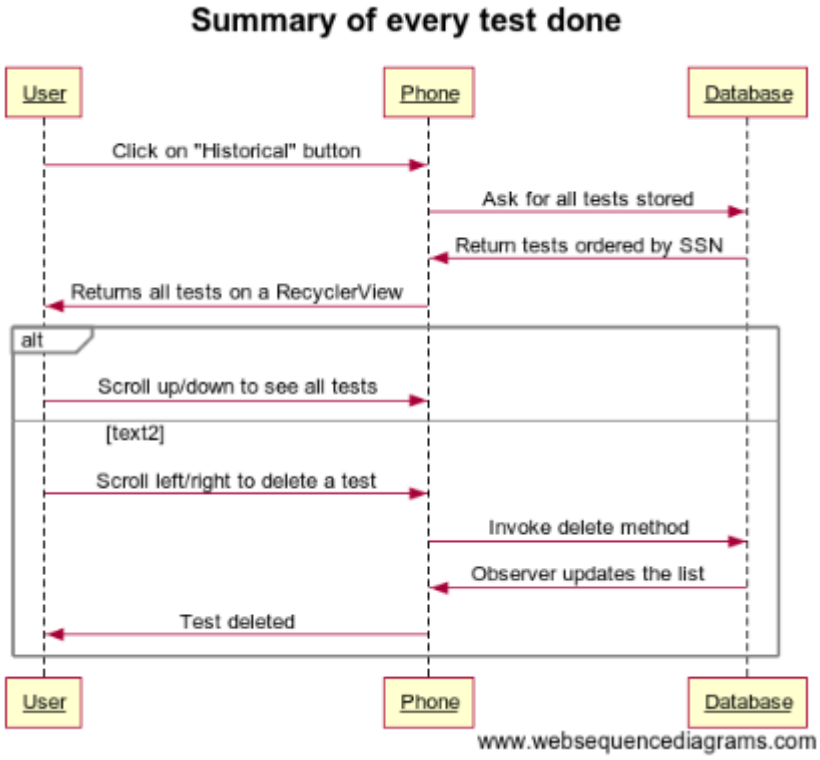


Figure 7 - Use-case 2: get a patient's score

This is the main purpose of the app, as we explained previously. The application needs to allow users to perform the test as quickly and as easily as possible, with few intuitive touches on the screen. As illustrated in the sequence diagram, as soon as the user starts one of the applications, it only needs one button to begin the test. Then, it only needs to input the information through the keyboard (we will focus on this feature later in this chapter) and click on the "Next" button. It will get all the information needed displayed on the screen while the phone does all the work in the background, creating the NEWS2/TEWS item (this will also be a focus later on), displaying the score/warnings and managing the creation/insertion of items in the database.

Once the test is finished, insertion on the database will be triggered. We will discuss later how the database work.

Use-case 3: Get a summary of every test done



As this application will be working mainly as a digital notebook for the nurses, they must be able to retrieve the information stored on the database about every test that they made before. When they press the Historical button on the main screen, the application asks the database for a list of every test saved, and then displays it on a *RecyclerView* (see Term list). The items on this list are ordered by the unique ID for each patient (and then for the time that the test was done). This way the nurses are sure to get the information in the correct order.

If a user wishes to delete a test from the list, they only need to swipe the item in one of the vertical directions and the phone will ask the database to delete that item. The list will be automatically updated using *LiveData* and *RecyclerView*.

4.2 Architecture

By designing and using a structured architecture the application is both easier to code and easier to maintain. Using a modularized architecture allows any individual module implementation to be freely exchanged if the new implementation match the same patterns as the rest.

4.2.1 Internal architecture

We have decided on using a form of *Model - View - ViewModel* that is fronted as the best practice in the official Android architecture guide. (Google 2019, j)

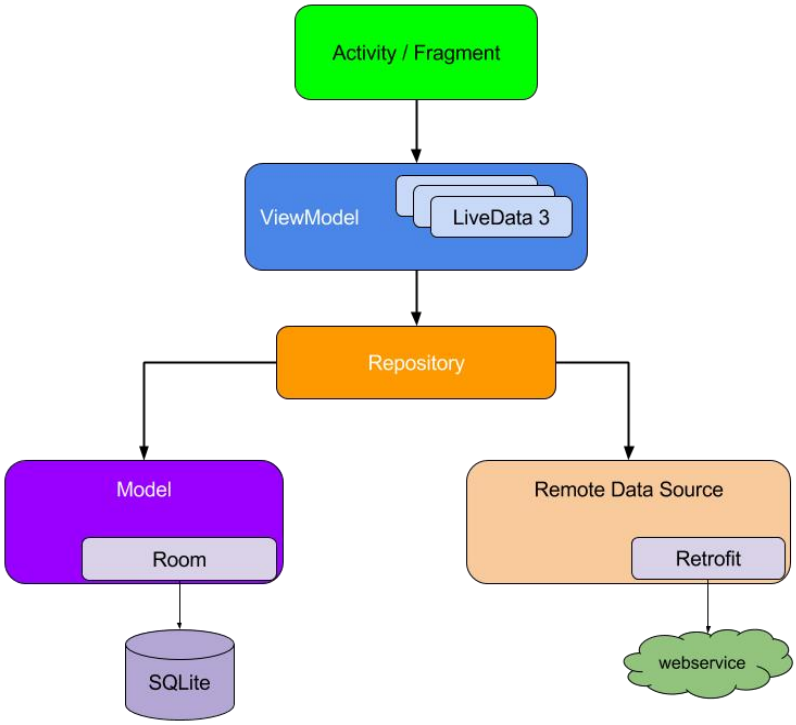


Figure 8 - MVVM architecture

4.2.1.1 Repository

A repository is a handler for gathering data for a system, such that code segments using the repository does not know where the data comes from, it simply trusts that it will get the data it requests from the repository, no matter the original source.

In the official Android application architecture guide it says:

> *"Repository modules handle data operations. They provide a clean API so that the rest of the app can retrieve this data easily. They know where to get the data from and what API calls to make when data is updated. You can consider repositories to*

be mediators between different data sources, such as persistent models, web services, and caches." (Google 2019, k)

This is very useful because it decouples the code such that you can change the repository to a different data source, and the rest of the code will function regardless. You may even have multiple data sources in the same repository. It is really an abstraction of the data sources, such that the code becomes more robust to changes to the system in future iterations.

4.2.1.2 Database

Our database consists of 3 entities represented by 3 tables with relations. One table for each application, TEWS and NEWS, containing a patient identifier, creation date and an identifying auto incrementing integer. Our database supports all CRUD operations in addition to special read queries through the Room framework, which is a standard Android framework for working with a SQLite database. We chose this framework as it is quite like other persistence libraries, such as for instance the Java Entity Framework or the .Net Entity Framework. It uses similar notation for primary keys, foreign keys and naming.

```
@Entity(tableName = "news2_table",
        foreignKeys = @ForeignKey(entity = Patient.class,
                parentColumns = "id",
                childColumns = "userId",
                onDelete = CASCADE,
                onUpdate = CASCADE),
        indices = {@Index("news2_ID"), @Index("creationdate")})

public class NEWS2 implements Serializable {

    /**
     * Automatically generated id for database purposes. For sec
     */
    @PrimaryKey(autoGenerate = true)
    private int news2_ID;

    /**...*/
    @NonNull
    private int userId;
```

Figure 9 - Example of the Room Framework Entity

This would make migrating the applications to another database environment and persistence library a much easier operation than using a hardcoded SQL-database.

Another benefit of using persistence libraries for handling database functionality is security. It helps mitigate risks posed by e.g. SQL injection attacks, denial of service attacks, minimize database load, and make data collection more efficient.

Persistence libraries also help prevent data leaks and corruption caused by several sessions trying to perform CRUD (create, read, update and delete) operations on the same entities by ensuring that the data is handled correctly and getting an acknowledgment of success after the operation is executed.

Our use of a repository also ensures asynchronous programming by implementing the AsyncTask interface of Android. This makes sure the application's use of data from various sources does not generate conflicts of interest if several persistence sessions are trying to access the same entity attributes and performing CRUD operations.

In conclusion we found the positive properties of using persistence libraries in conjunction with a repository so great that we implemented this.

However, we created a SQL database schema, just in case that some future student group wants to keep our work and integrate the application in a non local SQL environment.

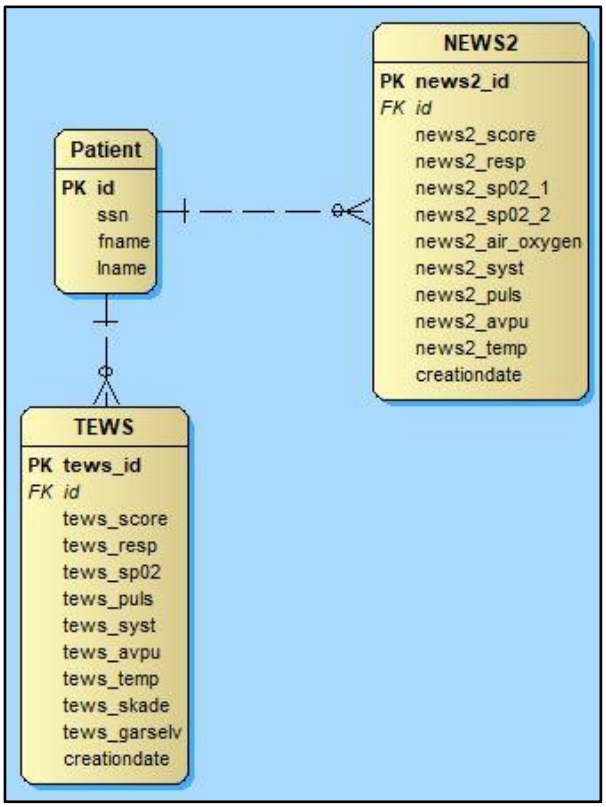


Figure 10 - E/R Diagram of the DB

```
/* ---------------------------------------------------------------------- */
/* Add table "NEWS2"                                                      */
/* ---------------------------------------------------------------------- */
CREATE TABLE `news2_table` (
      `id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
      `news2_ID` INTEGER NOT NULL,
      `news2_score` INTEGER NOT NULL,
      `news2_resp` INTEGER NOT NULL,
      `news2_sp02_1` INTEGER NOT NULL,
      `news2_sp02_2` INTEGER NOT NULL,
      `news2_air_oxygen` TEXT,
      `news2_syst` INTEGER NOT NULL,
      `news2_puls` INTEGER NOT NULL,
      `news2_AVPU` TEXT,
      `news2_temp` REAL NOT NULL,
```

```
      `timestamp` TEXT
);
/* ------------------------------------------------------------------ */
/* Add table "TEWS"                                                   */
/* ------------------------------------------------------------------ */
CREATE TABLE `tews_table` (
      `id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
      `tews_ID` INTEGER NOT NULL,
      `tews_score` INTEGER NOT NULL,
      `tews_resp` INTEGER NOT NULL,
      `tews_sp02` INTEGER NOT NULL,
      `tews_puls` INTEGER NOT NULL,
      `tews_systBT` INTEGER NOT NULL,
      `tews_AVPU` TEXT,
      `tews_temp` REAL NOT NULL,
      `tews_skade` TEXT,
      `tews_garSelv` TEXT,
      `timestamp` TEXT
);
```

This script defines the structure of the entities used by the application.

4.3 Modules

The following section contains a brief explanation of the internal design of the application.

4.3.1 Activities

Our applications are designed as a collection of ***activities*** (see *Term List*) that work together to offer the user the best User Experience (UX) possible. Each activity used can be seen on the images below.

Figure 11 - Brief look up at the Activities of each application

Every activity works almost the same way. We will discuss it deeply on the next section of the report, now it is just a brief introduction of concepts and the logic behind the functionality of the application.

```
public class TEWS_TemperatureActivity extends AppCompatActivity {
    private EditText et;
    private int intScore = 0;
    private boolean answered = false;
    private String log;
    private TEWS tews;

    @Override
    protected void onCreate(Bundle savedInstanceState) {...}

    public void checkInput() {...}

    public void next() {...}

    @Override
    public void onBackPressed() {...}
}
```

Figure 12 - Standard skeleton of an activity

- ***onCreate module***: creates the activity, binds the layout elements to the variables used and handles the input of data through keyboard.
- ***checkInput module:*** when the user inputs data, this module ensures that the value is correct (i.e. no weird or empty values) and handles the calculation of the total score, according to the value submitted by the user.
- ***next module:*** when the value has been successfully validated and processed, this module saves the instance of the TEWS2/NEWS bundle and sends it to the next activity
- ***onBackPressed module:*** if the user made a mistake by submitting a wrong value, this module allows it to go back to the previous activity and fix it. It restores the bundle value to the previous one so all the changes that the wrong value made to the bundle are denied.

The order of the activities is set according to the NEWS2 and TEWS spreadsheets provided and explained by the Bergen Municipality workers, and shown in Figure 2 (The TEWS scoring system) and Figure 3 (The NEWS2 scoring system)

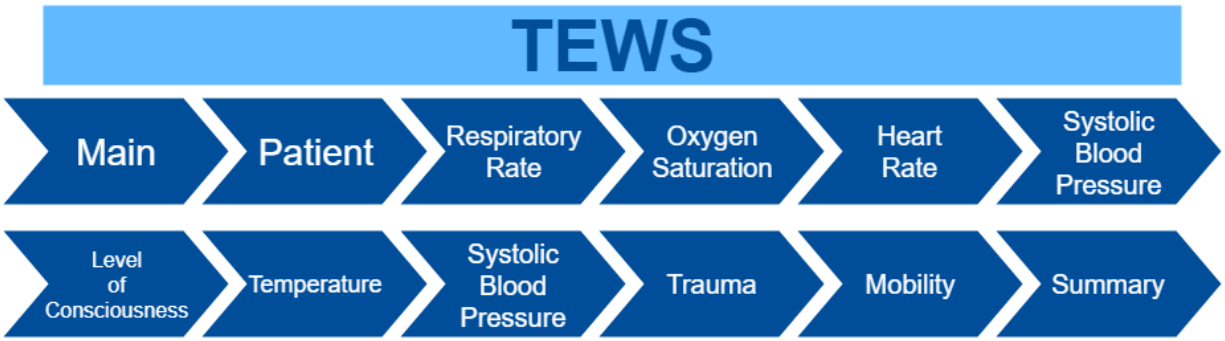


Figure 13 - Sequence of TEWS activities

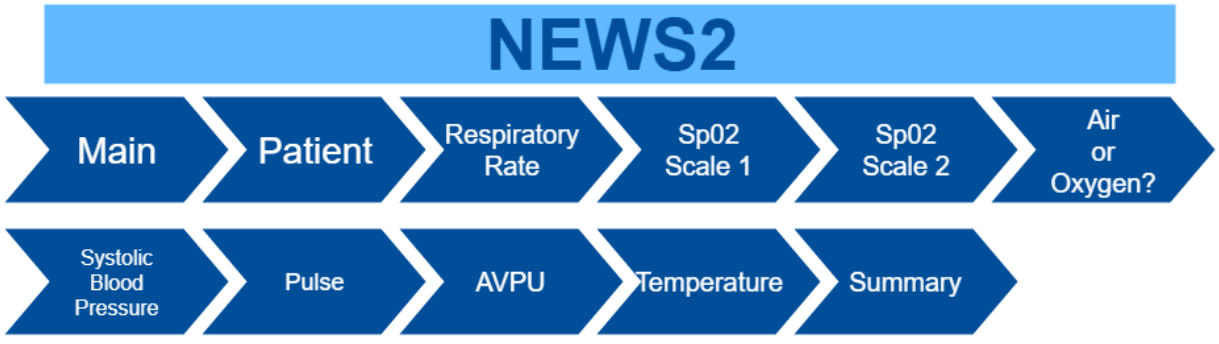


Figure 14 - Sequence of NEWS2 activities

4.3.2 Layouts

We declared our ***layouts*** as XML files, as shown below (see *Term List*)

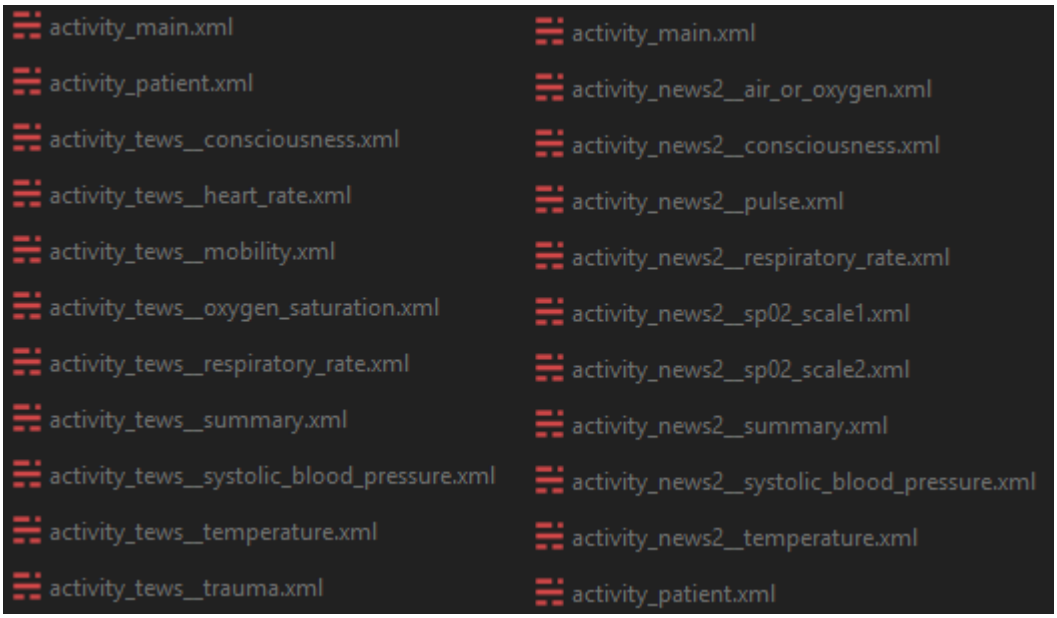


Figure 15 - Brief look up at the application layouts

As the activities, all the layouts are pretty much the same. All layouts are declared as ***Constraint Layouts*** (see *Term List*)

We chose Constraint Layout because it is the easiest (it is built basically by drag-and-drop elements into the screen) and most flexible type (it allows us to easily adapt our layouts to any kind of screen)

This is an example of one of the layouts. The blue lines indicate the “constraints” of one element, so it is designed to automatically adapt itself to the screen borders. This allows the application to work in almost every device while maintaining the screen ratios.

For the design pattern, we tried to stick to the main principles described by Google on ***Material Design*** (see *Term List*). We have been trying to use Material Design to provide users with an understandable experience by providing feedback to actions and changing the user interface to reflect changes the user is given information. We know our job is far from being perfect, we made a big progress in understanding what the client expects from us and what they want to see.

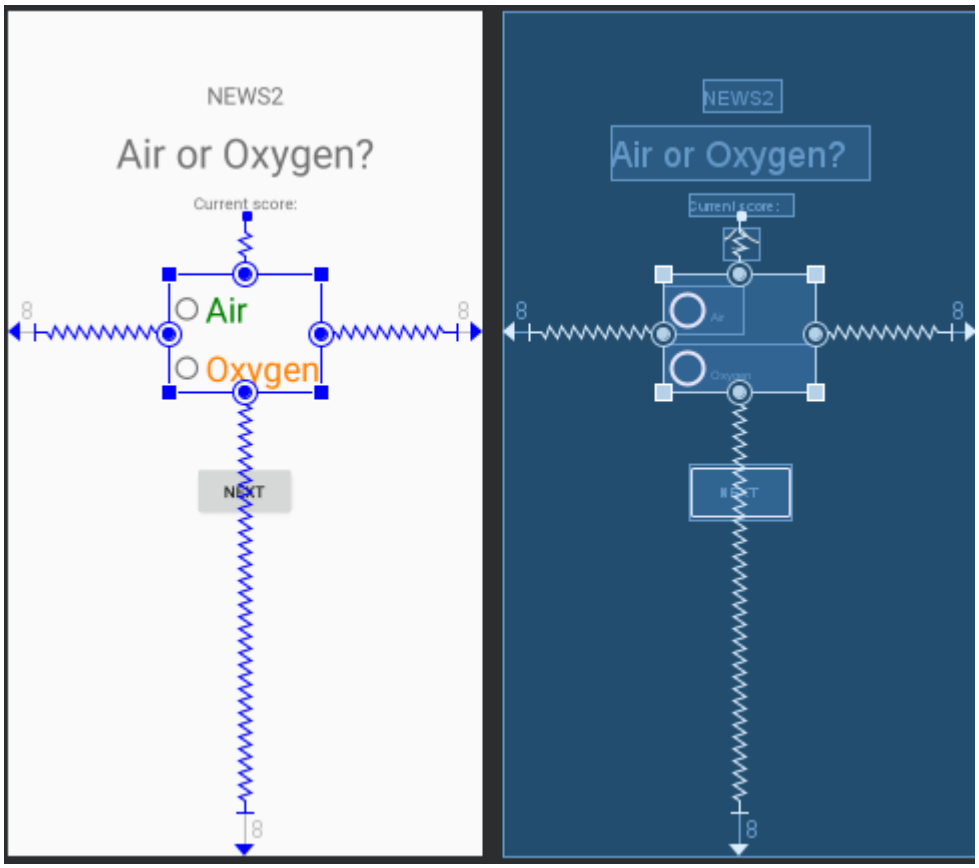


Figure 16 - How a layout is declared

4.3.3 String resources

We decided to use XML ***String resources*** (see *Term List*). This allows us to create many resources so we can easily translate our application to different languages. Obviously, we decided to create String resources for both English and Norwegian (both Nynorsk and Bokmål). So, according to the language of the device used, the app will automatically translate.

```
<string name="app_name">ProAct - NEWS2</string>
<string name="demo">NEWS2 DEMO</string>
<string name="news2">NEWS2</string>
<string name="begin">Begin test</string>
<string name="next">Next</string>
<string name="currentScore">Current score: </string>
<string name="totalScore">Total score: </string>
<string name="integerHint">0</string>
<string name="degreeHint">ºC</string>
<string name="percentageHint">0%</string>
<string name="patientInfo">Patient info</string>
<string name="patientName">Patient name</string>
<string name="patientID">Patient ID</string>
<string name="patientRoom">Patient room</string>
<string name="respiratoryRate">Respiratory Rate</string>
<string name="sp02_1">Sp02 Scale 1</string>
<string name="sp02_2">Sp02 Scale 2</string>
<string name="air_oxygen">Air or Oxygen?</string>
<string name="air">Air</string>
<string name="oxygen">Oxygen</string>
```

```
<string name="app_name">ProAct</string>
<string name="demo">Demo</string>
<string name="news2">NEWS2</string>
<string name="tews">TEWS</string>
<string name="begin">Start testing</string>
<string name="next">Neste</string>
<string name="currentScore">Foreløpig score:</string>
<string name="totalScore">Total score:</string>
<string name="integerHint">0</string>
<string name="degreeHint">0ºC</string>
<string name="patientInfo">Pasientinfo</string>
<string name="patientName">Pasientnavn</string>
<string name="patientID">Pasient-ID</string>
<string name="patientRoom">Pasientrom</string>
<string name="mobility">Mobilitet</string>
<string name="walking">Går</string>
<string name="withhelp">Med Hjelp</string>
<string name="immobile">Immobil</string>
<string name="consciousness">Våkenhetsgrad</string>
<string name="alert">Våken</string>
```

Figure 17 - Comparison between English and Norwegian string resources

4.4 Explanation of how the application works

When the user wants to use the application, first it needs to choose which one it needs

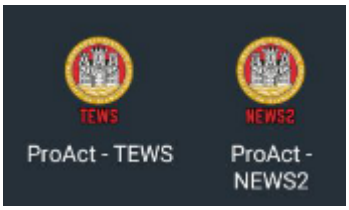


Figure 18 - Application icons

Once the application starts, the main screen will be shown to the user. As both applications are similar, we are only going to show screenshots of one of them.

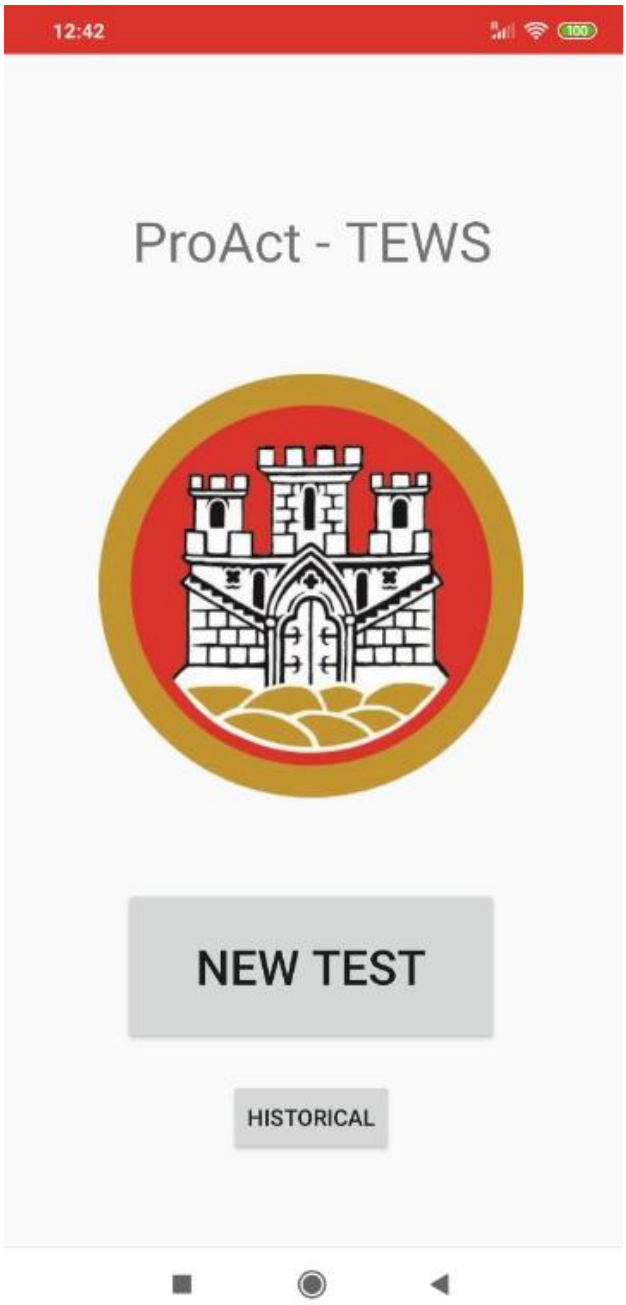


Figure 19 - Application main menu

This is the main screen of the application. By pressing the “New test” button the test will begin.

Figure 20 - Screen to input patient information

This is the first screen of the test itself. The keyboard opens automatically so the user only needs to worry about input the numbers needed.

Once the user pushes “Begin test”, a NEWS2/TEWS empty object will be created. The ID written by the patient will be stored at the object. Then, the object will be stored in a ***Bundle*** and sent through an ***Intent*** (see *Term List*) to the next activity.

Figure 21 - One of the screens to gather measurements

This is the standard screen for a test parameter. As you can see, it is quite minimalist but there is plenty of information displayed here:

- The green highlight shows which parameter needs to be input
- The red highlight shows the current score of the test. As it has only began, it is 0
- The orange highlight is the input field. The system automatically opens the keyboard, reducing the screen touches needed by 1. This requires the user to only input the value.

Once the user inputs the value, the application will make background checks (as explained on *4.3.1 Activities*). If the value is empty, a ***toast*** (see *Term List*) message will be displayed, alerting of the mistake.

If everything is successful, the TEWS/NEWS2 object will be updated, stored on the Bundle and sent to the next Activity through the Intent.

Figure 22 - Screen with Warning signal

The application loads the new activity. It reads the value of the Intent sent by the previous one and loads the information of the Bundle inside. As you can see:

- The green highlight has changed to a new needed value name
- The red highlight has automatically updated, according to the value of the previous input.
- The yellow highlight appears because the last value is dangerous. This allows the user to quickly notice it to quickly call a doctor.

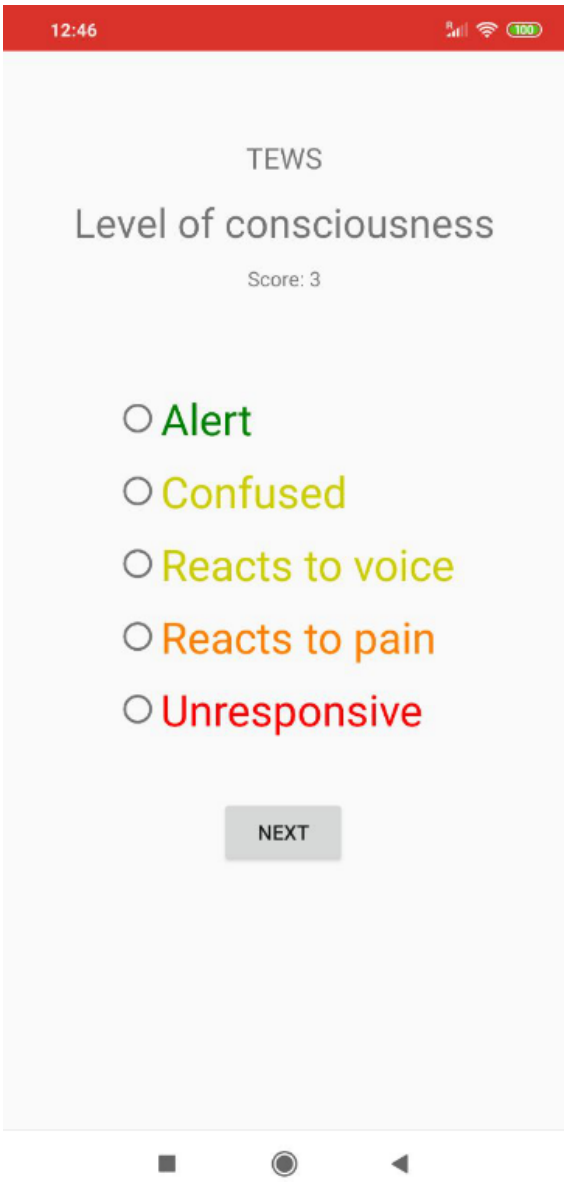


Figure 23 - Screen with radio buttons

After a few input screens, we get to see a different one, this time using a Radio Button. As you can see, the previous value is not considered dangerous, so the warning alert has disappeared. Now no keyboard is shown, the user is only required to select a button. Only one value can be selected, and if it tries to press “Next” without any value selected, a pop-up warning will be displayed.

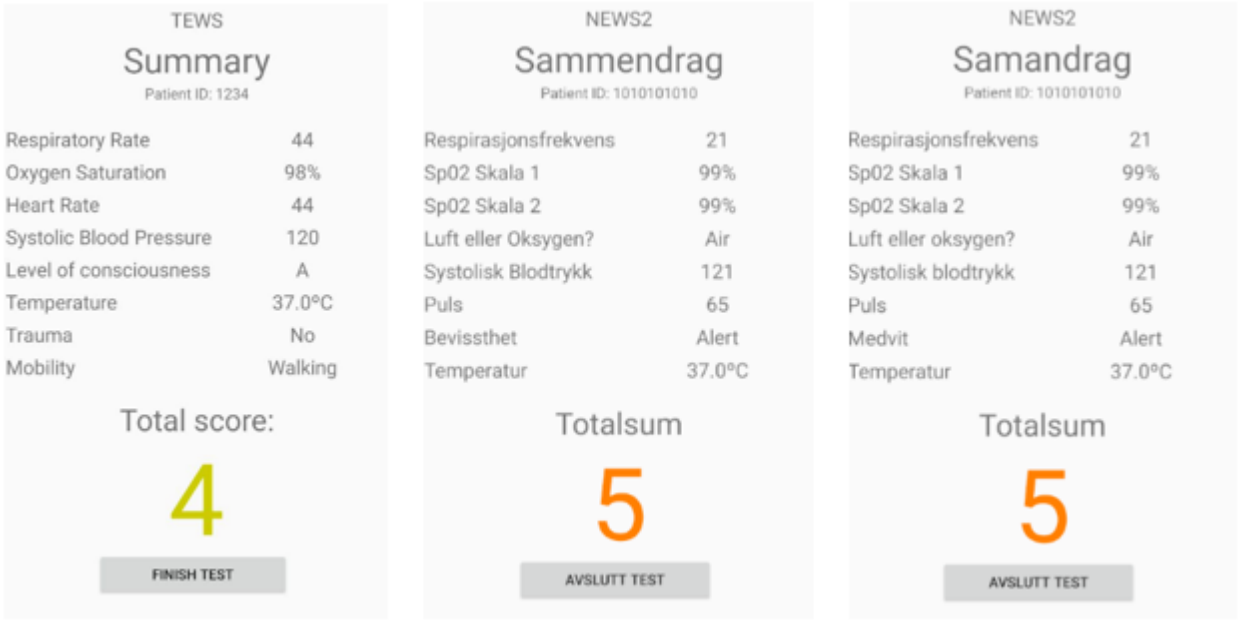


Figure 24 - Summary of the test in both English, Bokmål and Nynorsk

This is the last stage of the test. It displays all the information inside our Patient and TEWS/NEWS2 Bundle. The total score is displayed in different colours, depending on the severity of the patient (from best to worst: green - yellow - orange - red)

Once here, the TEWS/NEWS2 object is not stored on the database until the user press the “Finish test” button. This way, the user is still able to change a measurement. When the user presses the button, the object gets a timestamp from the device and it is stored on the database. The user is redirected to the main menu.

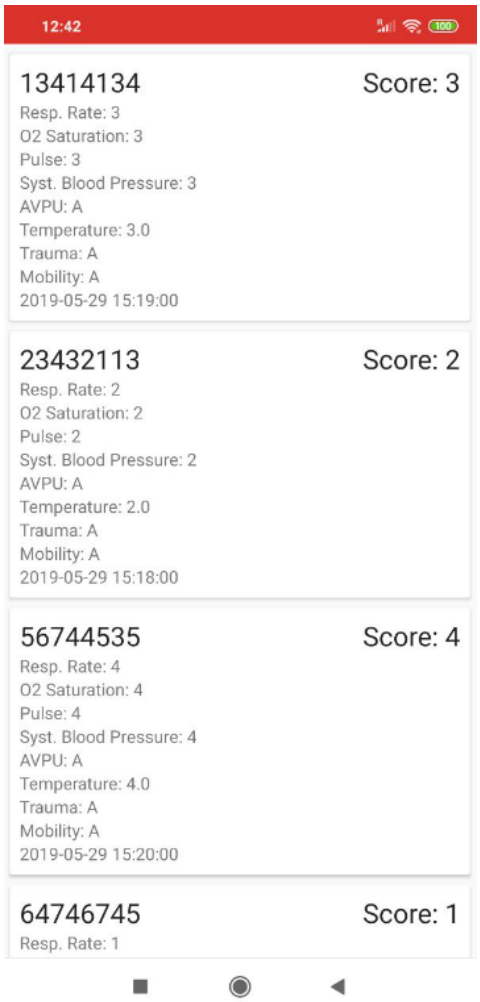


Figure 25 - Historical screen layout

When users press the "History" button on the main menu, they obtain a list of each TEWS / NEWS2 element stored in the database. Each item in the list shows the information of a test in a minimalist way, emphasizing the patient's ID and the score obtained. The order of the elements is determined by the ID.

When the information related to a test ceases to be relevant, the user can slide the item to the right or left to delete it from the database.

5. EVALUATION

Evaluation is a key part of the project as without evaluation it is impossible to properly determine whether the project complies with its specifications and if the project goals are met or not.

5.1 Evaluation methods

The project group used a few of the best practice feedback methods in order to get the best result possible with their current means. The following text will explore which processes that was decided and why they were chosen.

5.1.1 User feedback

The primary method for evaluating the result was using focus groups to deduce whether the project group had understood the purpose and use of the application, i.e. that the product was the correct application for the client.

This was accomplished by meeting with the client coordinator and 5 nurses of varying backgrounds to give the best feedback possible. Everyone met at the municipal hospital by the Bergen Accident and Emergency Department, the contact there was Kurt Arild Krokmyrdal, whose position there was as a clinical nurse specialist. A clinical nurse specialist decides what should be best practice for the other nurses on the workplace, and thus a great candidate for evaluating the development of NEWS2/TEWS applications.

It was decided that a sample group of 5 people would suit the project's needs best without seizing all the hospital's on-duty personnel.

At first, the focus group was asked for feedback on different use cases and a hand-drawn prototype and took notes. This was before there really was any code.

5.1.2 Team feedback

One of the best feedbacks you can get is a different worldview from a team developer. We went over each other's code continually to check if there was any better way to make the system.

5.2 Evaluation results

As stated previously in this report, the development of this project has revolved around achieving the best user experience and fulfilling all the requirements requested by the focus group.

Bergen Municipality nurses have been reviewing the application since its conception and provided feedback on the different iterations in development. Feedback was provided through several meetings with the focus group.

5.2.1 Meeting on 15.02.2019

This meeting was mentioned on *2.1.4 Initial solution idea.* It was useful to know the basics before getting started with the design of the application itself on Android Studio.

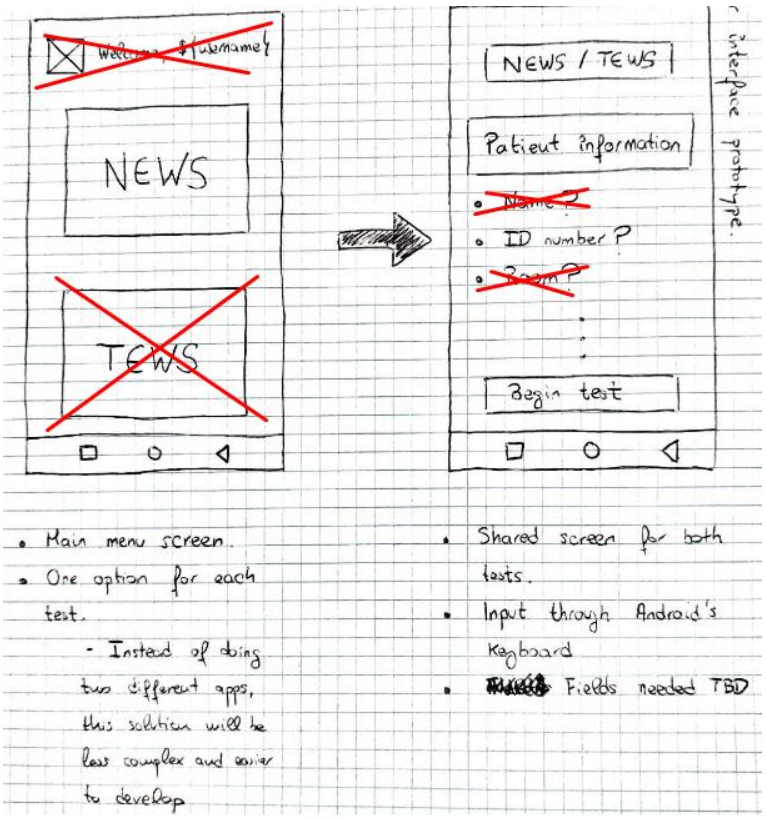


Figure 26 - Paper sketch with feedback from focus group (1)

As shown in *Figure 24*, the focus group agreed that they did not want any unnecessary information displayed on the screen. They also mentioned that they would prefer having two different applications instead of one with both tests. For the patient information, they said that a Norwegian ID number should be enough.

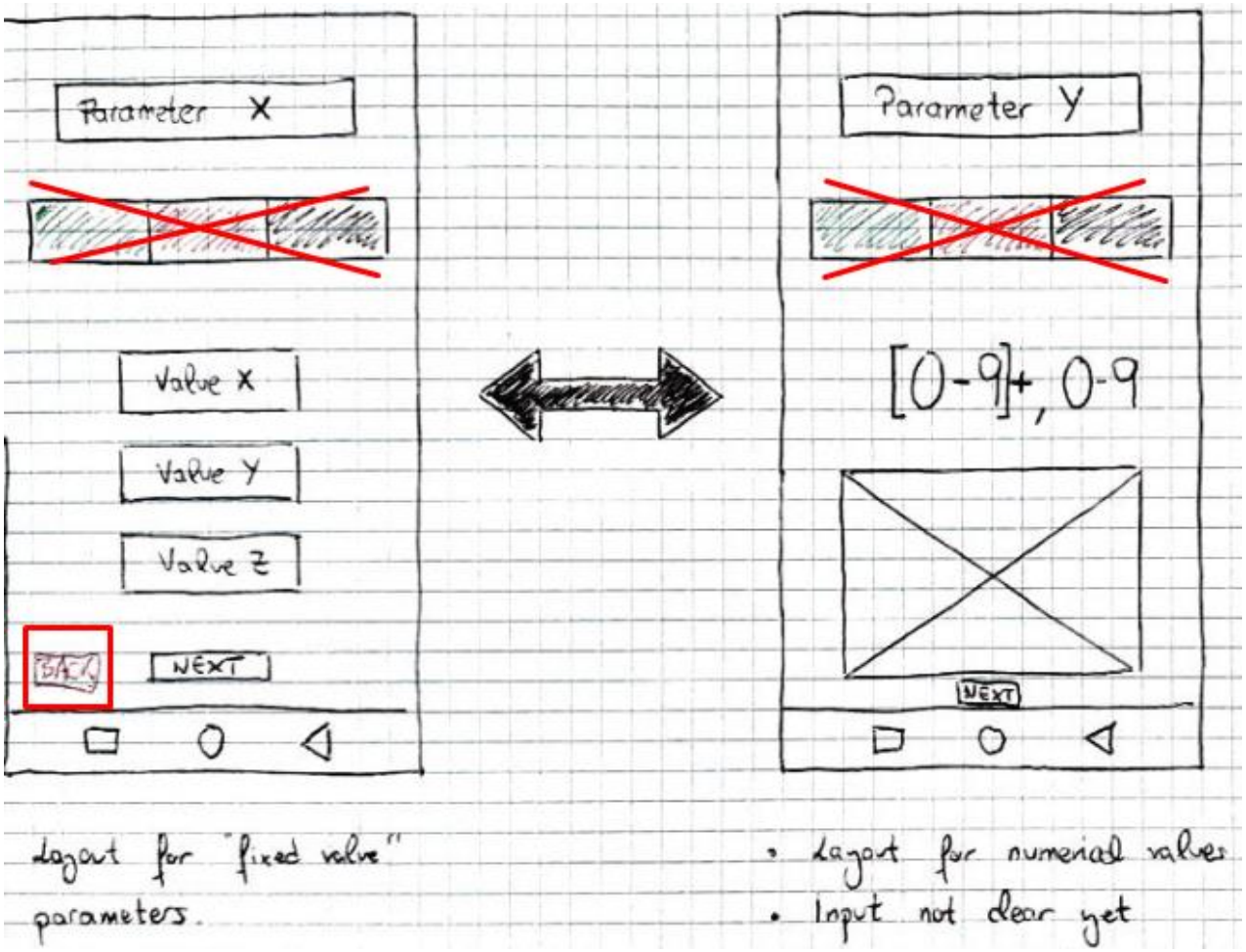


Figure 27 - Paper sketch with feedback from focus group (2)

To enter measurements, they thought that the sketch shown in *Figure 24* was fine. They agreed on the button layout for parameters like “Level of Consciousness” and the keyboard layout for numerical values. They also said that our sketch of a legend bar with information of the value ranges was redundant and unnecessary, as they only wanted the colours to be displayed on the total score. Additionally, they asked for:

- A “back” feature, just in case they input a wrong value by mistake
- A counter to display the total score in real time.
- A warning message that will be displayed if a value is considered dangerous for the patient.

5.2.2 Meeting on 09.05.2019

This meeting was held during the third iteration of development. We held one during the second stage, but that meeting was not attended by the focus group, therefore it was omitted as there was no new feedback.

For this meeting, the plan was to let the focus group try the application for the very first time and ask them to fulfil a brief questionnaire with their thoughts about it. They were able to use the application without having a previous explanation of how it works, so we think that was a big success regarding our design focused on user experience.

These were the questions asked on the questionnaire, from 0 being the worst score to 5 being the best. As you can see, we got a very positive feedback, that allowed us to close the UX iteration and focus on the backend developing.

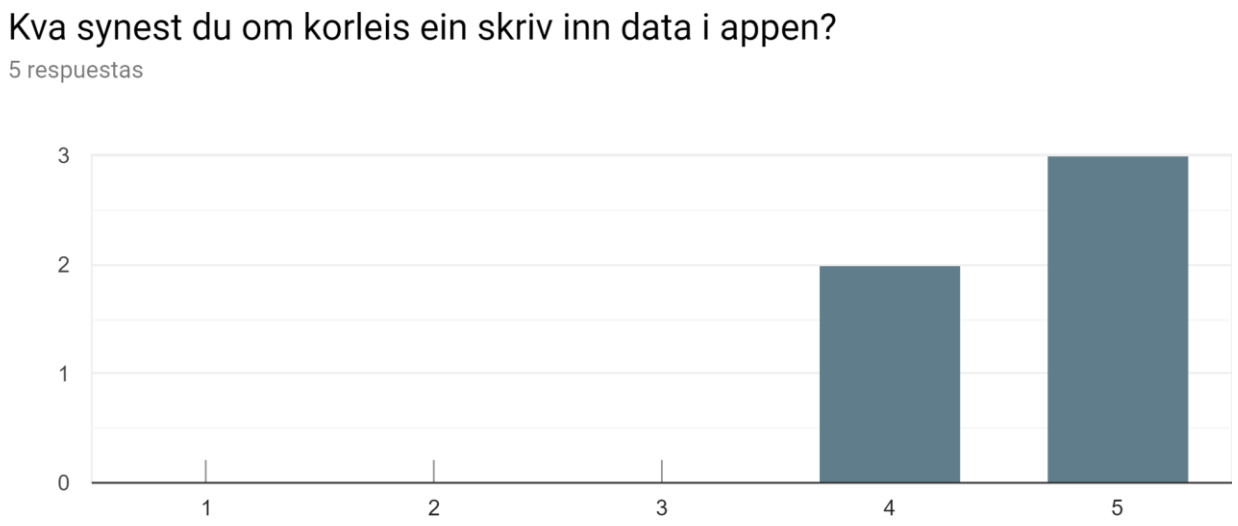


Figure 28 - Graph 1:

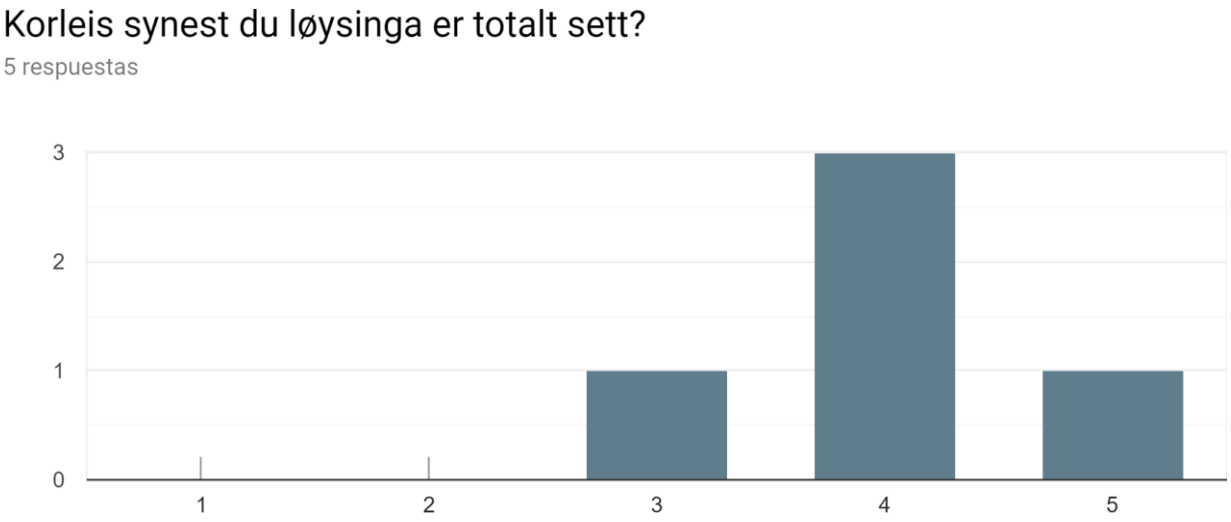


Figure 29 - Graph 2:

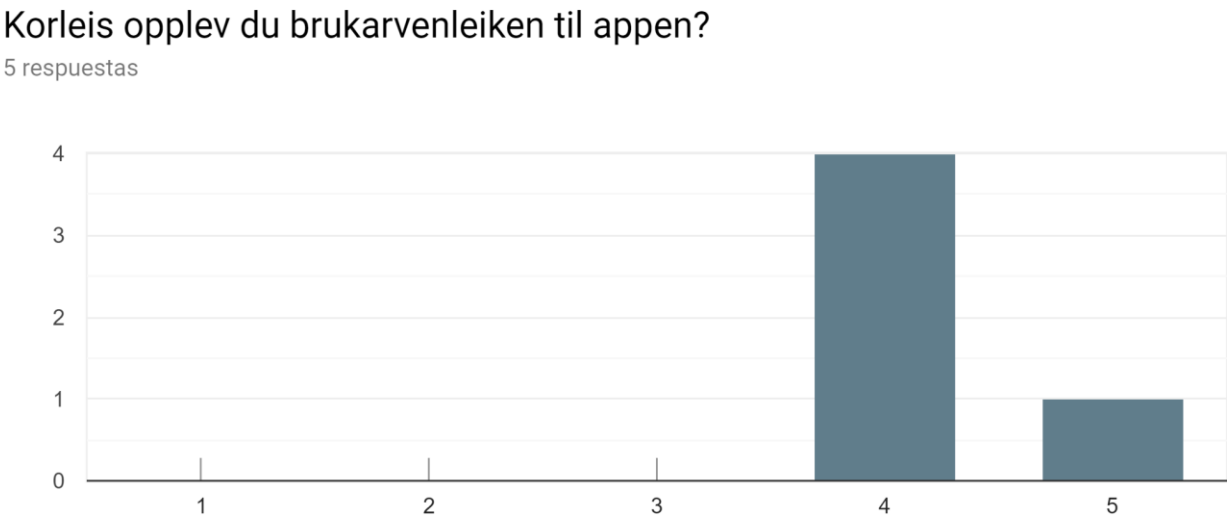


Figure 30 - Graph 3:

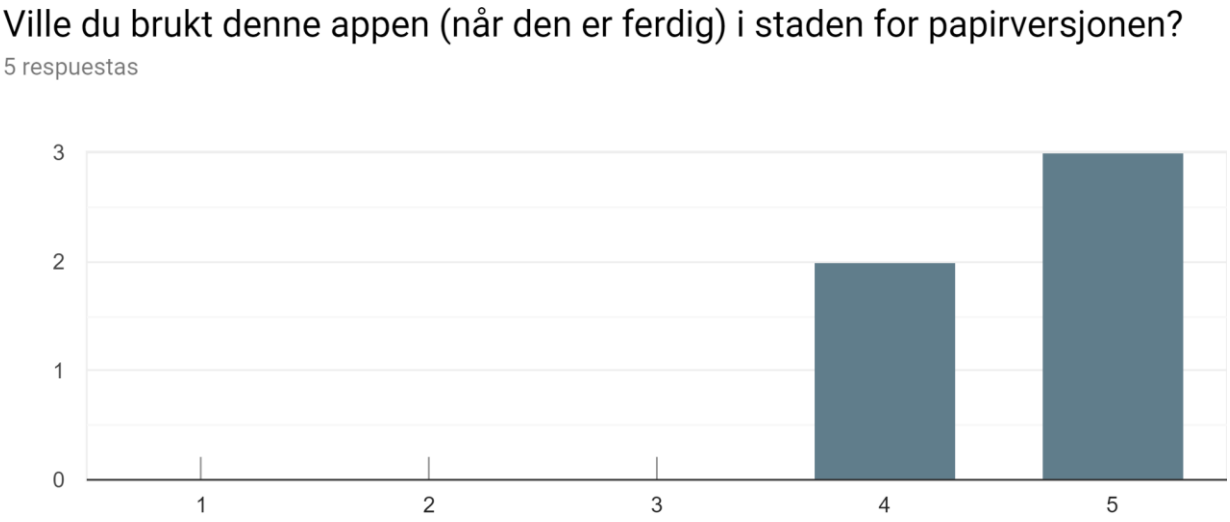


Figure 31 - Graph 4:

Figure 32 - Graph 5:

Figure 33 - Graph 6:

5.2.3 Meeting on 29.05.2019

This last meeting involved Kurt Arild Krokmyrdal, and the project group.

Since he was not present at the last meeting, he was shown all the progress made in the application. He provided positive feedback, stating that he liked how the application looked. He considered that it will be a useful notebook tool for nurses.

At the end of the meeting, a date was set for a last meeting, in which the future of the application will be decided, and the application will be distributed to the nurses if it is deemed useful.

6. RESULTS

Results show that the application is so far easy and intuitive to use, and by far much quicker and more efficient than the currently paper-based version.

The application does what it is supposed to do and the underlying mechanics work as intended. The feedback regarding user experience was good, so we think we achieved one of the main goals of this project.

The biggest issue so far is the database integration. Our solution works but it is far from being integrated on the Bergen Municipality data environment. However, we believe that we have created a good enough foundation for future students to address this problem.

In the backend, we have created the database structure. In the frontend we have created some solid documentation for best practice implementation according to a focus group trained as nurses. This will most likely prove invaluable in the future should some party decide to implement a similar solution on a larger scale.

The code produced follows best practices. It has a low coupling because no classes know what is happening directly above itself in the architecture, and a high cohesion because every class has well defined responsibilities and actions. The addition of a repository class added to this effect with the abstraction of data sources, as stated in previous segments.

7. DISCUSSION

7.1 Decision of approach

We only made a proof-of-concept application as we decided to not integrate into the production environment because of the timeframe of delivery.

Our architecture has been abstracted and can take data from any data source. In addition to being MVVM, we also have a repository module in the application that abstracts data from any source you deem sensible. In this case the data source is a SQLite database accessed through the Android Room persistence framework, but it can be rewritten to use anything from xml to web services or caches. This should make the integration into the municipal network more seamless.

7.2 Result

Our choice to use Android studio limited our capabilities to make cross platform solutions. However, Bergen Municipality currently only uses tablets and phones using the Android operating system. If that were an issue, we would have chosen a framework such as Xamarin, React Native, or similar. For our sake we chose Android Studio using Java as we both had common grounds with Java and Android Studio.

7.3 Improvements

7.3.1 Framework

We could have chosen another framework supporting cross platform development and/or deployment. Just because the municipality's standard operating system is Android for the time being, does not mean it will not change in the future. Also, cross platform development creates a wider target group if other clients than Bergen Municipality are interested in this work.

7.3.2 Database integration

When we first chose this project to work at, we did not know what kind of database architecture was being used by the Bergen Municipality. This was shown to the project group through a 45-minute-long presentation with a technician and the head of operations department. When the employer showed the database schema, we knew that we would not be able to manage all the implementation in only three months.

The network infrastructure of Bergen municipality was comprised of a symbiosis of municipally owned solutions and proprietary solutions, where the proprietary solutions were owned by private companies who had previously had Bergen municipality as a customer. We would have to ask for permission to use these solutions with the prospective owners, which they could decline.

We decided along with our coordinator that developing the application as a proof-of-concept for a local environment would be a good solution, but maybe with much less focus on the user experience it would have been possible to integrate the application on their environment.

Deprioritizing the user experience in order to implement the application in the production environment would defeat the purpose of the project. The purpose was for the nurses to use an application over the current paper-based sheet in order to make the data gathering more efficient and save the environment by reducing paper use.

7.3.3 Different screen sizes

As explained in 4.3.2 Layouts, the application was designed to work efficiently in a variety of screen sizes. However, currently the application has no explicit support of lower/higher resolution devices. It will still work fine, but there is much that can be done to properly support such devices.

This can be done by functional testing on different screen sizes and then changing the layout files for each resolution.

7.3.4 Testing

We had issues with time constraints and, since we are only two members, we could not dedicate the time and resources to perform automated tests for this project. Manual testing of changes was good enough for us, but it made it hard to detect unwanted side effects or said change. Thus, ideally, we should have made both unit tests for each method and integration test for various components.

We planned to use a mocking library called Mockito, a widely used tool for simulating system objects. This would let us test each component independently of each other.

8. CONCLUSIONS AND FURTHER WORK

This project set out trying to find a better way to perform TEWS and NEWS2 in a medical environment using a structured approach and an iterative development style. We addressed most of the objectives and the application is available form Bergen Municipality nurses.

8.1 Judgment

Results show that the application is so far easy and intuitive to use, and by far so much quicker and efficient than the currently used paper-based version.

The application functionally does what it is supposed to do and the underlying mechanics work. The feedback gotten regarding user experience is good, so we think we achieved one of the main goals of this project.

8.2 Further uses

Theoretically this application could be used by any health care organization that is familiarised with the TEWS/NEWS2 systems. As the application is open sourced under an MIT license this could be achieved by any programmer implementing their own version.

8.3 Further development

Further development can try to improve the user experience even further, but the main work should be done against the database integration, by connecting the application to a real SQL database environment.

We already provide a SQL model and script in this very report. We were not able to integrate our work into the Bergen Municipality database environment due to mainly time limitations. As it is a government structure, all the access to the databases must be encrypted and treated with extreme caution, and it was impossible to work on it while developing the implementation work of the application itself in just three months.

8.4 Closing notes

The author’s views concerning the product is that it has become something useful, however not what the intention was in the beginning. In the beginning the objective was to create a system that would completely replace the paper sheets the municipality currently uses and store the data on the municipal servers. It has instead become a useful calculator and notebook for everyday use in the municipal care ward or out in the municipal home care services rounds.

We can see it reducing paper usage completely as well as making the data gathering more efficient for the health care workers. This has been confirmed through the feedback given from the clinical nurse specialist on the final meeting before this report was finished.

If this will be used in production will remain to be seen.

Source code

NEWS2:
https://bitbucket.org/juariash/proact-news2/src/master/

TEWS:
https://bitbucket.org/juariash/proact-tews/src/master/

9. LITERATURE/REFERENCES

Google (2019 - a), Material Design Introduction [Internet]:

Available from: <https://material.io/design/introduction/ > [Read: March.2019]

Google (2019 - b), Activity [Internet]:

Available from: <https://developer.android.com/reference/android/app/Activity> [Read: March.2019]

Google (2019 - c), Layout [Internet]:

Available from: <https://developer.android.com/guide/topics/ui/declaring-layout> [Read: March.2019]

Google (2019 - d), Constraint Layout [Internet]:

Available from: <https://developer.android.com/training/constraint-layout> [Read: March.2019]

Google (2019 - e), Intent [Internet]:

Available from: <https://developer.android.com/reference/android/content/Intent

> [Read: March.2019]

Google (2019 - f), String resource [Internet]:

Available from: <https://developer.android.com/guide/topics/resources/string-resource#String> [Read: March.2019]

Google (2019 - g), Bundle [Internet]:

Available from: <https://developer.android.com/reference/android/os/Bundle.html> [Read: March.2019]

Google (2019 - h), Room Persistence Library [Internet]:

Available from: <https://developer.android.com/topic/libraries/architecture/room > [Read: March.2019]

Google (2019 - i), Android Studio IDE [Internet]:

Available from: <https://developer.android.com/studio> [Read: March.2019]

Google (2019 - j), Model-View-ViewModel Architecture (MVVM) [Internet]:

Available from: <https://developer.android.com/jetpack/docs/guide> [Read: May.2019]

Google (2019 – k), Toast [Internet]:

Available from: <https://developer.android.com/guide/topics/ui/notifiers/toasts> [Read: February.2019]

Google (2019 – l), RecyclerView [Internet]:

Available from: <https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView> [Read: May.2019]

Google (2019 – m), Entity [Internet]:

Available from: <https://developer.android.com/reference/android/arch/persistence/room/Entity.html> [Read: May.2019]

Google (2019 – n), Foreign Key [Internet]:

Available from: <https://developer.android.com/reference/android/arch/persistence/room/ForeignKey> [Read: May.2019]

Google (2019 – o), Primary Key [Internet]:

Available from: <https://developer.android.com/reference/android/arch/persistence/room/PrimaryKey.html > [Read: May.2019]

Royal College of Physicians (2017) National Early Warning Score 2 [Internet].

Available from: <https://www.rcplondon.ac.uk/projects/outputs/national-early-warning-score-news-2> [Read: February.2019]

A Wallis, P & Gottschalk, Sean & Wood, Darryl & Bruijns, Stevan & De Vries, Shaheem & Balfour, C. (2006). The Cape Triage Score - A triage system for South Africa. South African medical journal = Suid-Afrikaanse tydskrif vir geneeskunde. 96. 53-6. [Read: February.2019]

Balzer, S. (2005). Contracted Persistent Object Programming [Internet]. Swiss Federal Institute of Technology Zürich. Available at <https://www.cs.cmu.edu/~balzers/publications/contracted_persistent_object_programming.pdf> [Read: May.2019]

Relational Database (RDB) (n.d.), in: *Techopedia* [Internet]. Available at: <https://www.techopedia.com/definition/1234/relational-database-rdb > [May.2019].

10. APPENDIX

10.1 Risk list

No	Description	P	C	R	Measures.
1	Short timeframe. Inability to deliver by deadline.	2	4	**8**	Plan well, and work diligently.
3	Loss of content. Destroyed hardware or software.	1	5	**5**	Create backups and redundancy.
4	Lack of understanding	2	3	**6**	Focus on learning new techniques that are required to complete the project.
5	Creating the wrong application.	1	4	**4**	Hold meetings with the client focus group and have a constant dialogue with the client.

List of risks ordered by the severity of said risk, where the scale for P and C are from 1 to 5
P: Probability that a situation might occur.
C: The consequence of that situation occurring.
R: Calculated by multiplying P and C

10.2 GANTT diagram

Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
	Project plan	**111 days?**	**Fri 11/01/1**	**Sun 16/06/**		
	1 Project start	**42 days?**	**Fri 11/01/19**	**Mon 11/03/1**		
	1.1 Planning and information gatheri	42 days	Fri 11/01/19	Mon 11/03/1		
	1.2 Group registration	7 days	Fri 11/01/19	Mon 21/01/1		
	1.3 Project selection	3 days	Wed 23/01/1	Fri 25/01/19		
	1.4 Client startup meeting	1 day	Fri 08/02/19	Fri 08/02/19		
	1.5 Project website deployed	1 day	Wed 13/02/1	Wed 13/02/1		
	1.6 Onboarding and clarifications	1 day	Tue 19/02/19	Tue 19/02/19		
	1.7 Intermediary objective planning	14 days	Wed 20/02/1	Mon 11/03/1		
	1.8 Project kickoff	1 day	Mon 11/03/1	Mon 11/03/1		
	2 Project build phase	**55 days?**	**Tue 12/03/19**	**Sun 26/05/19**		
	2.1 Project title	1 day	Wed 20/03/1	Wed 20/03/1		
	2.2 Pre project report	1 day	Mon 01/04/1	Mon 01/04/1		
	2.3 Pre project presentations	2 days	Fri 05/04/19	Mon 08/04/1		
	2.4 Status report	1 day	Tue 30/04/19	Tue 30/04/19		
	2.5 Report draft	1 day	Tue 21/05/19	Tue 21/05/19		
	2.6 Iteration 1	19 days	Mon 18/03/1	Thu 11/04/19		
	2.7 Iteration 2	16 days	Thu 11/04/19	Thu 02/05/19		
	2.8 Iteration 3	13 days	Thu 02/05/19	Mon 20/05/1		
	2.9 Final build	6 days	Mon 20/05/1	Sun 26/05/19		
	3 Project finish	**6 days**	**Mon 27/05/1**	**Mon 03/06/1**		
	3.1 Report revision	1 day	Mon 27/05/1	Mon 27/05/1		
	3.2 Blog content	1 day	Mon 27/05/1	Mon 27/05/1		
	3.3 EXPO Poster completed	1 day	Fri 31/05/19	Fri 31/05/19		
	3.4 Final report deadline	1 day	Mon 03/06/1	Mon 03/06/1		
	4 Post report phase	**10 days**	**Tue 04/06/19**	**Sun 16/06/19**		
	4.1 Reflection notes	1 day	Tue 04/06/19	Tue 04/06/19		
	4.2 Presentations	2 days	Tue 11/06/19	Wed 12/06/1		
	4.3 EXPO	1 day	Thu 13/06/19	Thu 13/06/19