SINU 2.0

Project documentation

Student(s):

* Octavian-Mihai Matei

**Group: 30431**

**Contents**

[I Project specification 3](#_Toc97792711)

[1.1 Domain Model Diagram 4](#_Toc97792712)

[II Use-Case model 5](#_Toc97792713)

[2.1 Users and stakeholders 5](#_Toc97792714)

[2.2 Use-Case identification 5](#_Toc97792715)

[2.3 UML Use-Case diagram 8](#_Toc97792716)

[III Architectural design 10](#_Toc97792717)

[3.1 Conceptual architecture 10](#_Toc97792718)

[3.2 Package diagram 10](#_Toc97792719)

[3.3 Class diagram 10](#_Toc97792720)

[3.4 Database (E-R/Data model) diagram 11](#_Toc97792721)

[3.5 Sequence diagram 11](#_Toc97792722)

[3.6 Activity diagram 11](#_Toc97792723)

[IV Supplementary specifications 11](#_Toc97792724)

[4.1 Non-functional requirements 11](#_Toc97792725)

[4.2 Design constraints 11](#_Toc97792726)

[V Testing 12](#_Toc97792727)

[5.1 Testing methods/frameworks 12](#_Toc97792728)

[5.2 Future improvements 12](#_Toc97792729)

[VI Bibliography 12](#_Toc97792730)

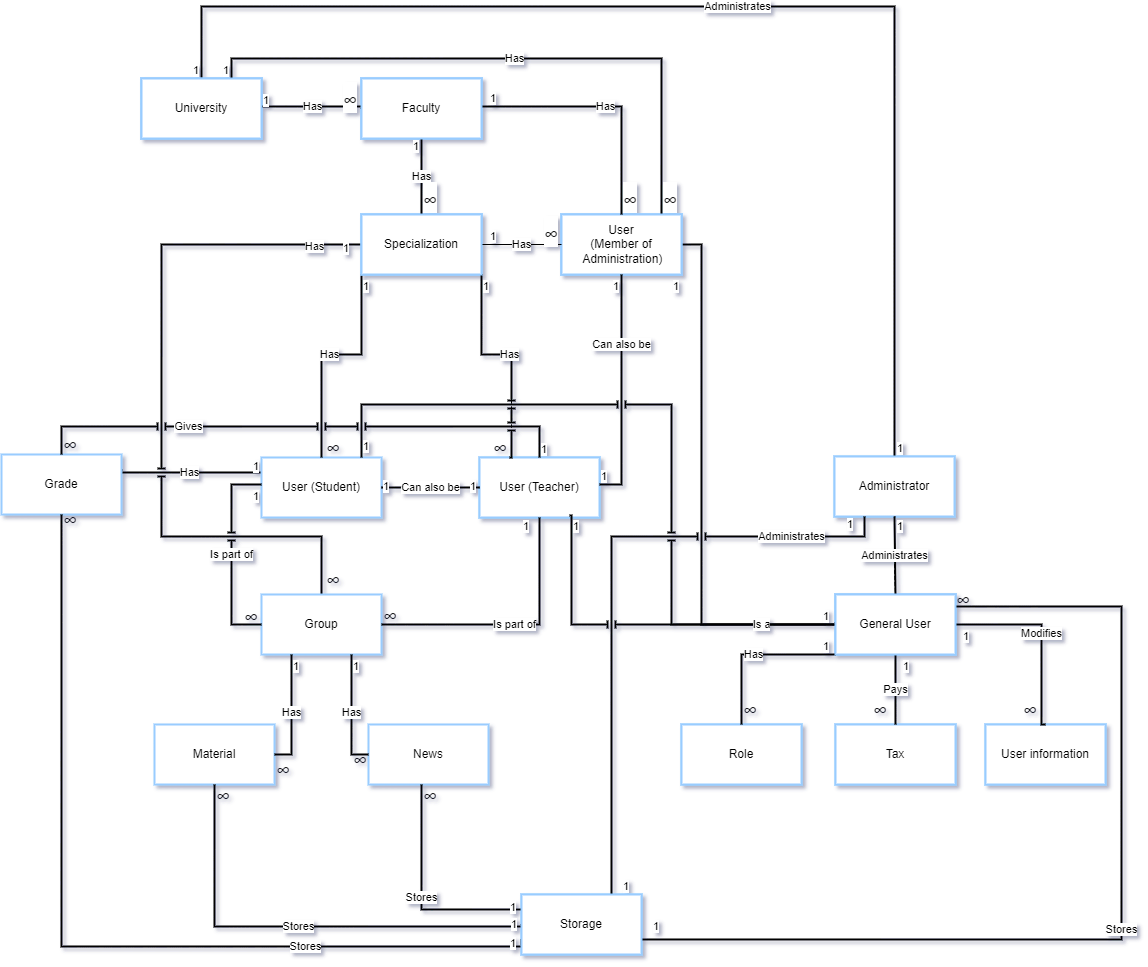
# I Project specification

SINU 2.0 proposes a revamp of the internal online management application used by TUC-N, [websinu.utcluj.ro](https://websinu.utcluj.ro).

This project was created in order to modernize and give a new face to the above application, having varied functionalities, such as tax payment, visualization of student groups, final grading and user information modification. Moreover, a great inspiration was the application [Moodle](https://moodle.org/?lang=eng), which has a grading system, but also a material support, in order to migrate everything to a single more manageable platform.

Also, the project will have three main user groups, the administration members, teachers and the students, so it can allow for a varied and exhaustive set of actions for all the roles. The system could also allow for a mix of these roles, in order to take into account the situations of administration-teacher or teacher-student.

## Domain Model Diagram



# II Use-Case model

The project will have three main user groups, the administration members, teachers and the students, so it can allow for a varied and exhaustive set of actions for all the roles. The system could also allow for a mix of these roles, in order to take into account the situations of administration-teacher or teacher-student. Another user would be the general admin, the one which administrates all the university, role given only to the university rector and possibly some administration members, in order to make a smooth continuity.

## 2.1 Users and stakeholders

* ***users***​
* Student
* Teacher
* University/Faculty Administration member
* Platform administrator
* ***stakeholders***​
* Project Manager/ Developing Team/ Manager - Octavian-Mihai Matei
* Project Manager/ Consultant - Alexandru Frasie
* Competition – Websinu, Moodle

## 2.2 Use-Case identification

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use case name:** Post news

**Level:** A user posts news on the news feed of the group

**Main actor:** Teacher

**Main success scenario:**

The teacher accesses the platform, then they login, access the group and choose the option to post news on the news feed of the group. They write the message, optionally upload a picture or/and document and then they post it. After this flow of actions, all the participants of the group should be able to visualize the new feed.

**Extension:**

When the teacher uploads the image and/or the document, the file could have the wrong extension or size, so the platform warns the user about this.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use case name:** Pay a tax

**Level:** A user pays a university tax

**Main actor:** General User

**Main success scenario:**

The user accesses the platform, logs into it and then accesses the taxes option from the menu. They are presented with a variety of tax options such as dorm payment, language test, parking on the university’s premises etc. They select the necessary options and then press the “Pay” button. The user inserts the card data and a successful message is displayed with confirmation on email.

**Extension:** NaN

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use case name:** Give grades

**Level:** A user gives grades to the students registered in the group

**Main actor:** Teacher

**Main success scenario:**

First, they will login, then access the respective group. After this, the teacher will register the grade for a specific student, with optional feedback option. These grade and feedback will be available for the student to see.

**Extension:**

If the student retakes the evaluation, then a modification or addition to the grade system can be made.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use case name:** Create new teacher account from existing account database

**Level:** An administration member user promotes a student account to a teacher account

**Main actor:** Member of administration

**Main success scenario:**

First, they will login, then access the management page of teachers and select to add another educator. Then they will search for the account’s email and name in order to find the user. After this, the account will have double role, one of a student and one of a teacher.

**Extension:**

If the email and name combination is wrong, an error will be displayed

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use case name:** Add users to group

**Level:** A teacher will add users with different roles

**Main actor:** Teacher

**Main success scenario:**

First, they will login, then access the group page. They will have the rights to modify the group, so they will access the addition portion of the page and select the email and the role of the new user in the group.

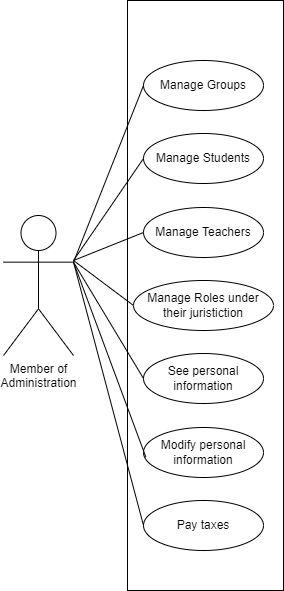
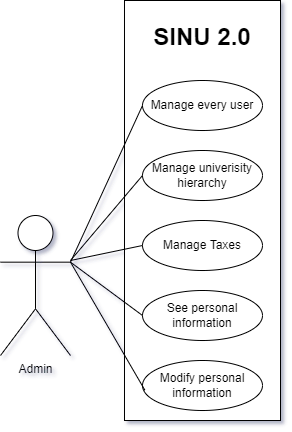
**Extension:**

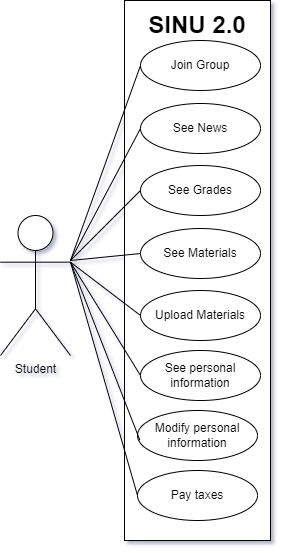
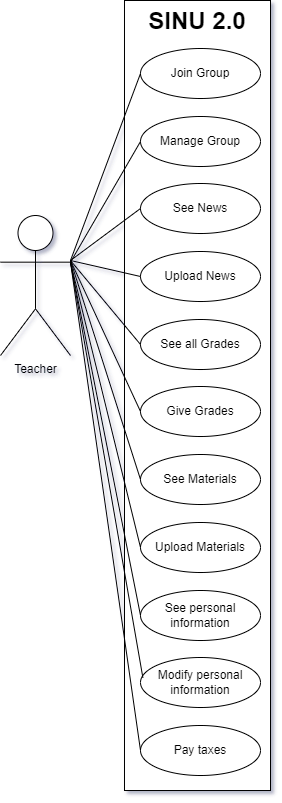
If the email and name combination is wrong, an error will be displayed.

The roles of the user can be:

* If the new user is only a student, then the role can only be student.
* If the new user is only a teacher, then the role can only be teacher.
* If the new user is both a student and a teacher, then the role can be either a teacher or a student.

## 2.3 UML Use-Case diagram



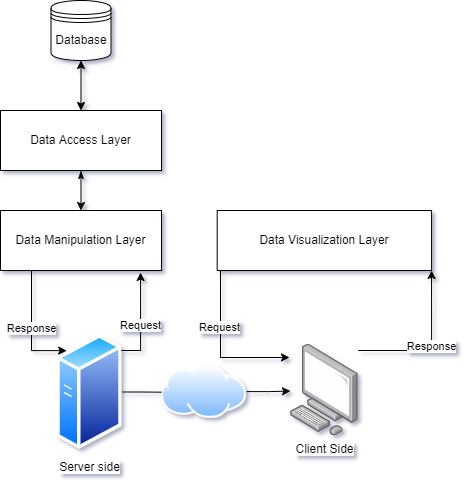
**

# III Architectural design

## 3.1 Conceptual architecture

The application is a web application using the framework ASP.NET, because it provides an exhaustive array of features and behaviors.

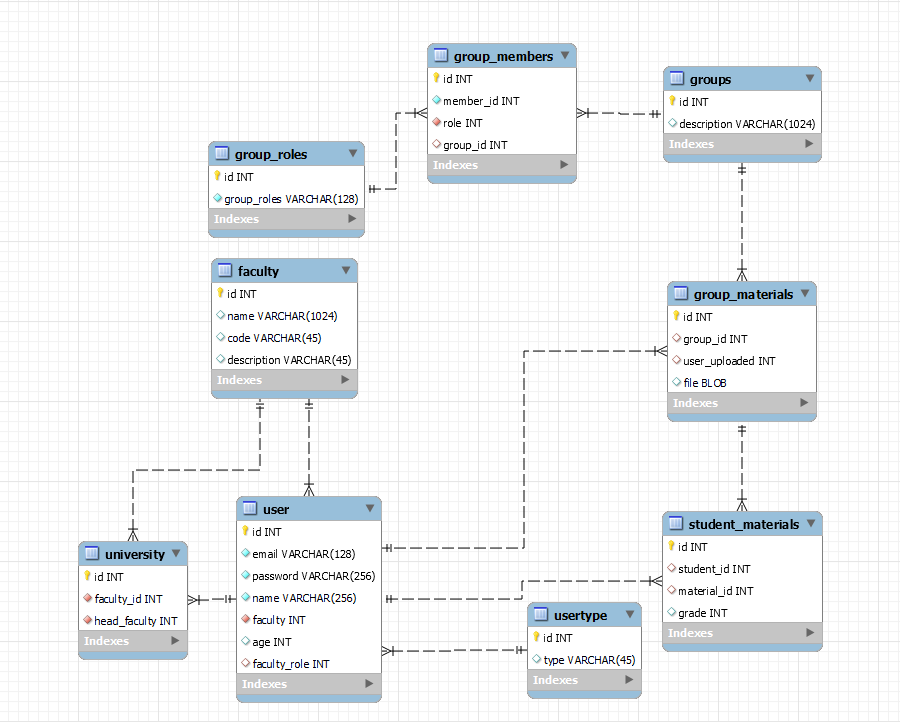
This allows for a Client-Server style application which is in conformity with the latest innovations in the field. The software is split between the Server side and the Client side, where the server provides REST calls to the client side. The server is split in three main parts: The Data Access Layer, which accesses the database, and the Data Manipulation Layer and the API interface which behaves, along side the Data manipulation Layer as a controller.



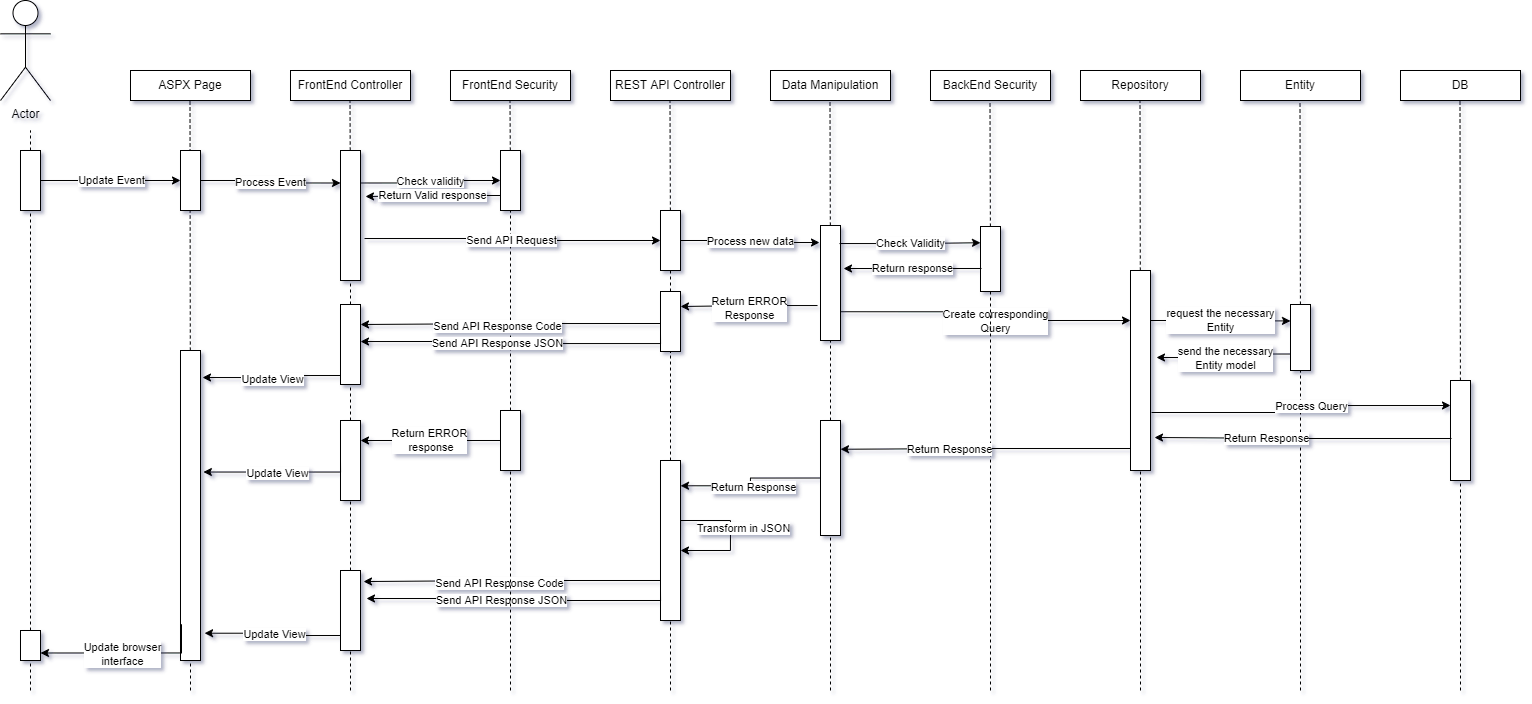
## 3.2 Package diagram

## 

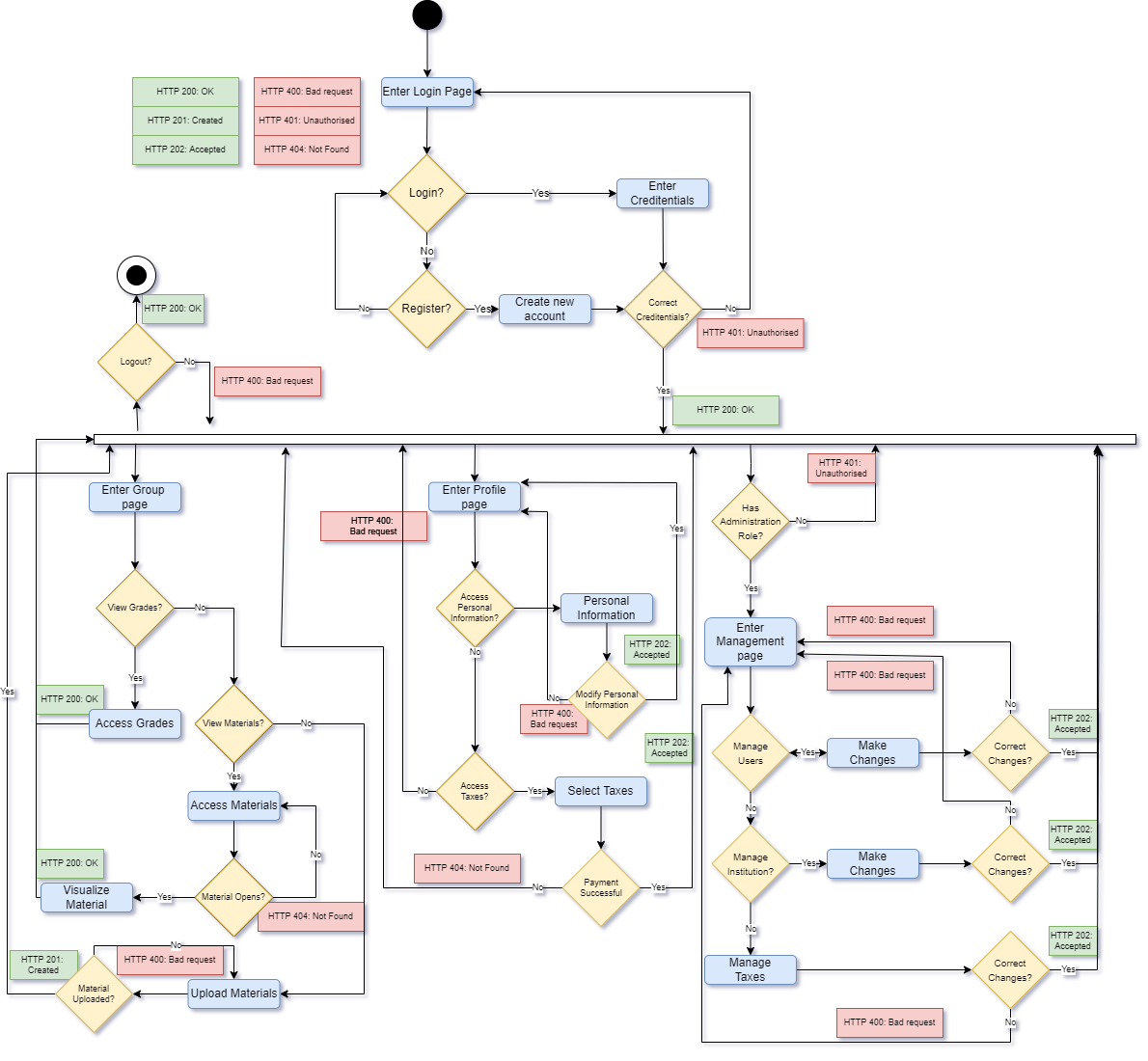
## 3.3 Database (E-R/Data model) diagram



## 3.4 Sequence diagram



## 3.5 Activity diagram



# IV Supplementary specifications

*< Se va scrie o mica introducere./>*

## 4.1 Non-functional requirements

1. Security

* The behavior of the software must be correct and predictable.
* The software must ensure the integrity of the customer account information.
* The server must not return a restricted web page to any browser that it cannot authenticate.
* The server must not return a restricted web page to a user who is not authorized to access it.
* The software must not accept overlong input data.
* The application must not accept invalid URLs.

1. Reliability

Measure degree to which the system performs as required:

* Includes resistance to failure
* Ability to perform a required function under stated conditions for a
* specified period of time
* Very important for critical, continuous, or scientific systems

Can be measured using:

* Probability that system will perform its required function for a specified interval under stated conditions
* Mean-time to failure
* Defect rate
* Degree of precision for computations

1. Capacity/Scalability

The capacity of a system refers to the amount of storage it offers. When using some applications, users can adjust and save settings based on their preferences. In this case, the user can store grades, materials, change personal settings etc.

1. Usability

* Navigation: When an application is usable, users can easily navigate its interface.
* Purpose of features: With high usability, users can easily determine what a feature is and what it can do.
* Quality of performance: When a device performs well, it means that the features of a system are functioning well based on what a developer predicted.

## 4.2 Design constraints

1. Framework – In order to implement the application, the main framework will be ASP.NET, which has a great variety of pre-built functionalities, but it can be hard to maintain and implement fast-type applications
2. Database – The database management and modification procedures could take a lot of time to implement and should have safeguards in order to facilitate a secure application to the user
3. Timescales – The implementation of the application will be in a quick timeframe, which could lead to hardcoded parts and semi-implemented features.
4. Testing – In order to test the application, a series of tests will be created. These will range from single use tests to exhaustive testing in order to test the capabilities of the application.
5. Bug fixing –This chapter will be a mix of manual testing and results from the unit tests implemented

# V Testing

*< Se va discuta la laborator./>*

## 5.1 Testing methods/frameworks

xUnit

## 5.2 Future improvements

# VI Bibliography