CASCHOM SCHOLARSHIP APPLICATION FOR CAMEROONIAN STUDENTS IN MOROCCO

Aboubakar Ibrahima
Supervised by: Naeem N. Sheikh

30th April 2019
Capstone Report

I, Aboubakar Ibrahima, affirm that I have applied ethics to the design process and in the selection of the final proposed design. In addition, I have held the safety of the public to be of paramount importance which has been in the presented design wherever may be applicable.

__________________________________________
Aboubakar Ibrahima

Approved by the Supervisor

__________________________________________
Dr. Naeem N. Sheikh
Table of Contents

ACKNOWLEDGEMENTS .................................................................................................................. 5
LIST OF FIGURES .......................................................................................................................... 6
ABSTRACT ......................................................................................................................................... 7
CHAPTER I: INTRODUCTION ........................................................................................................... 8
CHAPTER II: SOFTWARE METHODOLOGY ....................................................................................... 9
CHAPTER III: SYSTEM REQUIREMENTS .......................................................................................... 10
  1. Description of the Current Process ......................................................................................... 10
  2. Functional Requirements ...................................................................................................... 11
  3. Non-Functional Requirements .............................................................................................. 11
      A. Availability ......................................................................................................................... 11
      B. Usability ............................................................................................................................ 12
      C. Reliability .......................................................................................................................... 12
      D. Performance ..................................................................................................................... 12
      E. Confidentiality .................................................................................................................. 12
      F. Scalability ......................................................................................................................... 12
  4. Use Case Diagram .................................................................................................................. 12
  5. Class Diagram ....................................................................................................................... 13
  6. Entity Relationship Diagram ................................................................................................. 14
CHAPTER IV: TECHNOLOGY ENABLERS ....................................................................................... 15
  1. Node.js .................................................................................................................................... 15
  2. Nodemailer: ............................................................................................................................ 15
  3. MySQL .................................................................................................................................... 16
  4. Visual Studio .......................................................................................................................... 16
CHAPTER V: IMPLEMENTATION DETAILS .................................................................................... 18
  1. Model View Controller (MVC) .............................................................................................. 18
  2. Schedule .................................................................................................................................. 19
  3. Some Decisions ....................................................................................................................... 19
CHAPTER VI: RESULTS OF IMPLEMENTATION .............................................................................. 21
  1. Student .................................................................................................................................... 21
  2. MINESUP ............................................................................................................................... 23
  3. MINFI ...................................................................................................................................... 25
  4. AMBACAM .............................................................................................................................. 26
CHAPTER VII: STEEPLE STUDY ...................................................................................................... 28
  A. Societal ..................................................................................................................................... 28
  B. Technological ......................................................................................................................... 28
C. Environmental ................................................................. 28
D. Ethical........................................................................... 28
E. Political............................................................................ 28
F. Legal ............................................................................... 28
G. Economic....................................................................... 29

CHAPTER VIII: FINAL REMARKS......................................................... 30
1. Challenges and limitations ......................................................... 30
2. Future Works...................................................................... 30
3. Conclusion......................................................................... 31

REFERENCES........................................................................ 32
ACKNOWLEDGEMENTS

The accomplishment of this project would not be possible without many individuals. Firstly, I would like to thank my supervisor Dr. Naeem N. Sheikh for his significant and helpful suggestions in planning and developing this work. His willingness to give his time so generously was greatly appreciated.

Secondly, special thanks goes to my friends whose words comforted me when I felt overwhelmed by this project.

Lastly, I would like to thank my family for their love and unconditional support, not only during my studies but also throughout my life.
LIST OF FIGURES

Figure 1: Use Case Diagram of CASCHOM
Figure 2: Class Diagram of CASCHOM
Figure 3: ERD of CASCHOM
Figure 4: MVC of CASCHOM
Figure 5: Creation of a folder from student’s profile
Figure 6: Main page from student’s profile
Figure 7: Submission List from MINESUP’s profile
Figure 8: Rejection Box from MINESUP’s profile
Figure 9: Archive List from MINESUP’s profile
Figure 10: Submission List from MINFI’s profile
Figure 11: Main page from AMBACAM’s profile
Figure 12: Send an email to all button from AMBACAM’s profile
ABSTRACT

CASCHOM is a web development application made especially for Cameroonian Students in Morocco. The purpose of the application is to facilitate the renewal of their scholarship and also to get their funds in a shorter time. In fact, there is three documents that should be sent for that purpose. Within the application, they will be able to upload those documents via their account, keep track of each step of the process by receiving emails after validation or rejection (from the submission of the documents to the Ministry of Higher Education in Cameroon until the reception of their fund at the Embassy in Rabat), see the history of the previous uploaded documents.

CASCHOM allows students to re-upload documents easily in case of rejection.
CHAPTER I: INTRODUCTION

The number of Cameroonian students receiving scholarships in Morocco keep increasing every year because the Cameroonian government and the Kingdom of Morocco have a partnership allowing Cameroon to send students to different cities of Morocco regardless to their major. Today, there are about one thousand and five hundred (1,500) Cameroonian Students in Morocco who receive scholarships every six months. The renewal scholarship’s procedure is as follows: Firstly, a list of documents must be sent to the Cameroonian Students Association in Morocco (CASAM); Secondly, these documents are sent to MINSUP through google drive link for validation; Thirdly, a list of valid students is sent to Ministry of Finance (MINFI); Fourthly, embassy of Cameroon in Morocco (AMBACAM) receives the fund and finally, students can stop by AMBACAM to get their funds.

It looks like a simple task but this process takes many months to be completed. Hence, to alleviate the delays in the accessibility of funds by students, “CASCHOM” would shorten this process while facilitating the renewal of Cameroonian Students’ scholarships who would obtain their funds in a shorter period of time.
CHAPTER II: SOFTWARE METHODOLOGY

CASCHOM was developed using the incremental method. The first step was to gather information from the scholars and the process manager at the Cameroonian Embassy in Rabat. The collection of information in the initial step was not straightforward but my role as the vice-president of CASAM already gave me an idea on what was slowing down the process. Thus, I decided to carry out a survey which outcomes helped me to further my knowledge in the functional and non-functional requirements of the application. More importantly, this survey highlighted the pitfalls of this process.

After gathering all the requirements, the next step was to ensure that the project was technically feasible and economically viable through feasibility study and the STEEPLE analysis.

The design phase was done using the 3-Tiers architecture (client, server, and data side). The implementation phase took a while because I needed to get familiar with Node.js. I made sure to meet all the user requirements.

As of today, the testing and integration processes, which are done by my supervisor and myself, are still in progress. Once deployed, it will be carried out by volunteer scholars and other various parties involved in the process, namely the Ministry of Higher Education of Cameroon, the Ministry of Finance of Cameroon and the Embassy of Cameroon in Rabat. Their feedback will help me to improve the application for better usage. Maintenance will be done by improving the application and correcting any bugs.
CHAPTER III: SYSTEM REQUIREMENTS

1. Description of the Current Process

The current renewal scholarship’s procedure is done in six (6) steps as follows:

Step 1: A list of documents must be provided including the enrollment certificate of the current year, AMCI Scholarship Certificate, and MINESUP scholarship certificate, to the Cameroonian Students Association in Morocco who will in turn hand them over to MINESUP through a google drive link.

Step 2: MINESUP will check if the upload documents are valid then the name of valid students will be added to a list which they will forward to MINFI. The list of all the students having an issue will be sent to CASAM for resubmission.

Step 3: MINFI will validate the list sent by MINESUP and then transfer the funds according to the number of students.

Step 4: Once AMBACAM receive the funds, they let CASAM know.

Step 5: CASAM make a post on their official Facebook page and send emails to different branches of the association.

Step 6: Finally, students can stop by AMBACAM to get their funds.

The problem of this process starts after collecting all students’ files. In fact, the google drive link is sent by email within a department at MINSUP. From there, it should be forwarded to responsible individual. However, people working at MINSUP do not check their emails often times. In addition, even when CASAM decides to keep in touch with these people, they do not pick up the phone coming from an unknown number. This lack of professionalism results in a long period of waiting. Luckily, few files which move forward in the process still take several months to be screened. These files are then sent to MINFI by email later in the process.

Once again, CASAM still has to contact the responsible individuals at MINFI who will act the way as their compatriots at MINSUP (They do not check their emails or pick up the phone...
coming from unknown numbers). After several months when MINFI has done screening the files, the money can finally be sent to AMBACAM.

The full scholarship’s process takes several months up to years. As of today, students are still waiting for their funds which were supposed to be disbursed since July 2018.

2. Functional Requirements

- Each scholar must have an account created by the administrator of the application. The user’s profile contains: last name, first name, email, department, country and city.

- Every student can add, delete, and edit uploaded documents. Edit works only before submission.

- Fund confirmation is done through students profile as well as AMBACAM profile. The time and date of the transaction is recorded.

- MINESUP is able to validate or reject one or more uploaded documents. In case of reject, the reasons will be sent to the student by email. Furthermore, the reasons will also be archived on MINESUP profile in case the student appeals.

- MINFI will receive the list of students who are in good standing for validation.

Once AMBACAM receives the fund, they can send an email to all students and should be able to confirm that a student has taken the fund.

3. Non-Functional Requirements

A. Availability

The application is available anywhere at any time using any type of mobile device. The only condition is that, the user must have an internet connection.
B. Usability

The application must be user-friendly. For instance, if the user enters incorrect credentials, a message should appear and should ask for the correct ones. In case an individual forgets the login information, it can be sent by email.

C. Reliability

Bugs must be minimal so that users can enjoy the application.

D. Performance

The application should have a fast response time. The quicker, the better.

E. Confidentiality

Every scholar should only see what is on their profile.

F. Scalability

Regardless of the number of people connected at the same time, the application must maintain the same performance for each person.

4. Use Case Diagram

A use case diagram is a great way to communicate complex ideas in a fairly basic way. It points out the primary elements (actors) and processes (use cases) that form the system. In other words, it is a representation of a user's interaction with the system.
5. Class Diagram

A class diagram is a sort of static structure graph that describes the structure of a system by showing the relationships and multiplicity among objects, the system's classes, operations, and their attributes.

It classifies the actors defined in the Use Case Diagram into a set of interrelated classes.

---

**Figure 1:** Use Case Diagram of CASCHOM

**Figure 2:** Class Diagram of CASCHOM
6. Entity Relationship Diagram

It is a diagram that shows different tables of the database and relationships between them.

Figure 3: ERD of CASCHOM
CHAPTER IV: TECHNOLOGY ENABLERS

1. Node.js

According to www.tutorialspoint.com, “Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices”.

Released for the first time on the 27th of May 2009 by Ryan Dahl, it is written in C, C++ and JavaScript. It can execute JavaScript code outside of a browser. It supports several operating systems including Linux, Microsoft Windows, and macOS.

I decided to use Node.js because it is the technology that is gaining momentum nowadays. In contrast with ordinary web response where the client always initiates communication, with Node.js both the client and server can start the communication. This allow exchange of information in a quicker manner.

2. Nodemailer:

This module for Node.js was developed to allow easy email sending by Andris Reinman in 2010. At that time, there was no easy tool for emailing within Node.js. Now, Nodemailer has become very common tool for this task. All the reference I consulted on usage of Nodemailer
came directly from documentation on the Nodemailer website maintained by Reinman (www.nodemailer.com).

3. MySQL
MySQL is a free software and relational database management system written in C and C++. It is developed by Oracle Corporation and it was first released on the 23rd of May 1995. I decided to use MySQL because it is known for being the most secure and dependable database management system used in famous web applications such as Facebook. Besides, it is the most used database in the academic field. The one I learned in the database class was SQL which is pretty much something very similar..

4. Visual Studio
Developed by Microsoft Corporation, Visual Studio is a free software code editor. It is used for creating and debugging modern web and cloud applications. It was first released on the 29th of April 2015.
I could even use Sublime Text as a text editor for my script files, but I made the choice to use Visual Studio because it is flexible and indicates syntax errors of Node.js code.
CHAPTER V: IMPLEMENTATION DETAILS

1. Model View Controller (MVC)

MVC is an application design model that has three interconnected parts: the model (data), the view (user interface), and the controller (processes that handle input).

![THREE-TIER ARCHITECTURE]

Figure 4: MVC of CASCHOM

Client tier (also known as presentation tier) is the I/O part. In other words, the user enters some inputs and expects some output. It contains web browsers which could be on cellphones, computers or any other type of mobile devices.

Application tier (also known as Business logic tier) is the business part. As such, the part controls and works the whole process. It contains all of the code that would process. I used as programming language Node.js (JavaScript) on server side and HTML, CSS on the client side and finally visual studio code for the implementation.
Database tier (also known as database server) is the part that stores and manages data. It contains all the files and the database management system that would contain data on any actions taken from each user. I used MySQL as database.

2. Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project selection</td>
</tr>
<tr>
<td>2-3</td>
<td>Initial specification</td>
</tr>
<tr>
<td>4-6</td>
<td>Feasibility study and analysis</td>
</tr>
<tr>
<td>7-10</td>
<td>Design of the system</td>
</tr>
<tr>
<td>10-14</td>
<td>Development of the application</td>
</tr>
<tr>
<td>15</td>
<td>Final report</td>
</tr>
<tr>
<td>15</td>
<td>Capstone defense</td>
</tr>
<tr>
<td>15</td>
<td>Update final report</td>
</tr>
</tbody>
</table>

3. Some Decisions

Firstly, make students upload MINESUP certificate only once every four years since the document is valid for that period of time. It will be useless to make them upload it every year.

Secondly, the fund confirmation button is added on the student as well as AMBACAM profile. That way, students cannot deny getting their funds because the time and date of the transaction are recorded on the database.

Thirdly, the reasons why MINESUP rejects a file is archived and send to the concerned student by email.
Lastly, after each submission by students, a notification is sent by email to the specific responsible of scholars of Morocco at MINESUP. In other words, when student X submit a file, instead of sending the email at the department in charge of scholars, I send the email to Y who is charge of scholars in Morocco.

Email sending

```javascript
function sendMailto(email, messages) {
  var mailOptions = {
    from: 'appcrawler1@gmail.com', // sender address
    to: email, // receiver address
    subject: 'CASCOM HAIL', // subject line
    text: messages,
  };
  transporter.sendMail(mailOptions, function(error, info) {
    if (error) {
      console.log(error);
    } else {
      console.log('Email sent!');
    }
  });
}
```

```javascript
if (roles.role.code == 'MINESUP') {
    madame['etats'] = 3;
    testeur = true;
    sendMailto('mini190@gmail.com', "Sir/Madam, There are some documents that need validation on CASCOM platform. Your files have been validated by the Ministry of Higher Education and the Ministry of Finance."
    sendMailto(result[i].utilisateur.login, "Your files have been validated by the Ministry of Higher Education and the Ministry of Finance."
} else if (roles.role.code == 'MINFI') {
    console.log('------------------------------------------- 3');
    madame['etats'] = 3;
    testeur = true;
    sendMailto('ambassade90@gmail.com', "Sir/Madam, There are some documents that need validation on CASCOM platform. Your files have been validated by the Ministry of Higher Education and the Ministry of Finance."
} else if (roles.role.code == 'AMBASS') {
    console.log('------------------------------------------- 3');
    madame['etats'] = 3;
    testeur = true;
    sendMailto('mini190@gmail.com', "Sir/Madam, The Embassy of Cameroon in Rabat confirms that " + connected.nom +
```

Size and extension limitation

```javascript
if (this.files[0].size > 2097152) {
  alert("The maximum size allowed is 2 Mo + this.files[0].size");
  document.getElementById('btn-test').style.display = "none";
} else {
  var nonfichier = this.files[0].name;
  var result = ";
  var jk =
  var ik = nonfichier.length
  if (result == "pdf") {
    //alert("Only pdf documents are allowed!!") + document.getElementById('idmessages3').innerHTML;
    document.getElementById('btn-test').style.display = "none";
    try {
      document.getElementById('idmessages3').innerHTML = "Only pdf documents are allowed!!!</p>";
    } catch (r) {
      alert(r);
    }
  } else {
    // alert("Your document is uploaded!!!") =
    document.getElementById('btn-test').style.display = "block";
  }
```

20
CHAPTER VI: RESULTS OF IMPLEMENTATION

This is the main login page where different users (student, MINSUP, MINFI, AMBACAM, admin) can login to their respected page.

1. Student

A. Students can create a folder to upload the required documents.

- There is a size limitation: the application accepts a maximum of 2Mb documents. If students try to upload a bigger document, a message will appear to let them know that is impossible. In addition, the reminder about the size will be pointed out.

- There is an extension restriction: The application accepts only PDF documents. In case students try to upload another extension such as docx (word), txt (text) a message will appear to let them know that only PDF documents are allowed. Students will not be able to click the “save” button.
B. Students can add/edit/submit documents. They can also confirm receiving their funds.

- The “add” button is here in case the three documents are not available when the student login to their profile. In that case, students can upload what they have, save them and complete later by clicking on the “add” button.

- The “edit” button is available in case students want to make some changes. On the other hand, as soon as they submit they will not be able to edit anymore.

- The use of the “submit” button is to send the documents to MINESUP and at the same time send an email to the responsible individual at the same ministry so the person can log in, check the documents and validate them.

- After getting the fund, students have to confirm from their profile by clicking on “fund confirmation” button.

**Figure 5:** Creation of a folder from student’s profile

**Figure 6:** Main page from student’s profile
2. MINESUP

A. Responsible individuals at MINESUP can view the documents that were submitted, valid or reject them.

- After log in to the profile, these individuals can click on the submission lists. There, they can find all students that submit their three documents. They can see: student’s name and first name, the subject and the message. In addition, they can open (or download) the submitted documents.

- In case the three documents are valid, the file is validated by clicking on the “accept” button.

- In case one of the documents is not valid, the file is rejected by clicking on the “reject” button.

![Submission List from MINESUP’s profile](image)

**Figure 7:** Submission List from MINESUP’s profile

B. Rejection box.

- In case of rejection, the responsible individual enumerates the reasons on the rejection box. The reasons will be sent to the concerned student by Email.
C. In case of rejection, the reasons are archived.

- The reasons of rejection will be archived in case the student appeals. Student can re-submit the documents after fixing the issues.
3. MINFI

A. Responsible individuals at MINFI view and valid the list sent by MINESUP.

- After log in to the profile, these individuals can click on the submission lists. There they can find the list of students that had valid documents. The list include: student’s name and first name, the subject and the message.

- As soon as the responsible individual validates, the list is sent to AMBACAM.

- The only reason for which MINFI reject the file is MINESUP contact them after sending the list for any reason.

![Figure 10: Submission list from MINFI’s profile](image)
4. AMBACAM

A. Responsible individuals at AMBACAM view and valid the list sent by MINFI.

- As soon as the fund is sent, the responsible individual receives an email from MINFI with an estimation arrival date.

- After log in to the profile, these individuals can click on the submission lists. There they can find the list of students that MINFI validated from the list received from MINESUP. The list include: student’s name and first name, the subject and the message.

- AMBACAM confirms that students get the fund by clicking on “accept” button.

![Figure 11: Main page from AMBACAM’s profile]

B. AMBACAM send email to all valid students after receiving money

- When AMBACAM receives the money, they have to notify students by clicking on “send an email to all” button. An email will be sent to students that MINFI validated.
Figure 12: Send an email to all button from AMBACAM’s profile
CHAPTER VII: STEEPLE STUDY

A. Societal

CASCHOM is interested in the social issue by helping students to receive their funds in a shorter period of time. In fact, anyone with an account can log in using any type of mobile device, anywhere and at any time. As a result, managers (the people in charge of scholars) in various departments related to the provision of scholarships will be able to continue their duties, even at home, if necessary, without having to carry any documents.

B. Technological

CASCHOM helped me to use MySQL as a database, Node.js programming language on server-side and HTML, CSS on the client side, and finally Visual Studio Code for implementation.

C. Environmental

The application has no impact on the environment.

D. Ethical

From an ethical point of view, students will have full access to the manuscript which would guide them in the whole process. If there is a problem, they will know exactly where the process stopped or had issues. In addition, the application will help to avoid ethical irregularities such as corruption.

E. Political

The application has no political impact.

F. Legal

Free open sources were used for the creation of this application.
G. Economic

To create this application, it did not involve any cost due to the use of open source software. Nevertheless, it took me time to learn the programming language, Node.js, and thus deepen my knowledge of programming. Once the website is launched, Cameroonian Students Association in Morocco will be responsible for advertising CASCHOM through different channels such as their official Facebook page. The application will be free of charge.
CHAPTER VIII: FINAL REMARKS

1. Challenges and limitations

I faced some issue while working on this project. It was very difficult:

- To connect Node.js to Gmail: in fact, I had to read from different forums to overcome the issue.
- To have size restriction on the student’s profile (2Mb).
- To have extension restriction on the student’s profile (PDF only).
- To make sure students cannot edit any documents after submission.
- To archive the reasons why a file is rejected by MINSUP.

2. Future Works

Future work could include:

- Adding mobile payment: students will not travel to Rabat to get their funds.
- Making the mobile version of the application: students will be able to use the application from their phone.
- Adding multiple languages: Cameroon is a bilingual country (French and English). The current application is in English. Adding at least French will be interesting.
- Posting new scholarships along with requirements: scholarships along with requirements will be posted on the main page. Students that fulfill the requirements can then apply and follow up their application from the platform.
- Allowing students to apply for scholarships and keep tracking of their application: the application is now made only for renewal of scholarship, I will extend the application
to start from applying for the scholarship. Students will be able to check the status of their application.

- Archiving reasons why MINFI reject documents: the reasons given by MINESUP to reject the file after their validation should be archived.

3. Conclusion

In conclusion, what motivated me to work particularly on this project is the desire to provide a solution that will help my country mates. The use of the application has been applied to the case of Morocco in this project but in the long run, the goal is to include every country where Cameroonians scholars are present such as Russia, Malaysia and the United States of America. I want to emphasize that this work is by no means exhaustive and can be improved.
REFERENCES
Awad Ben. How to Send an Email in Node.js. September 9, 2017
  https://www.youtube.com/watch?v=YU3qstG74nw
Kif-Dev. NodeJS - Send free emails using Nodemailer and Gmail. Jan 17, 2017
  https://www.youtube.com/watch?v=JJ44WA_eV8E
Oracle. MySQL Tutorial.nd.
  https://downloads.mysql.com/docs/mysql-tutorial-excerpt-5.5-en.pdf
  https://www.softwaretestingmaterial.com/software-architecture/
Reinman. A. Message Configuration. nd.
  https://nodemailer.com/message/
Reinman, A. Nodemailer. nd.
  https://nodemailer.com/about/
Reinman, A. Usage. nd.
  https://nodemailer.com/usage/
Singh Negi Manoj. Sending an email using NodeMailer & Gmail. August 1, 2017.
  https://codeburst.io/sending-an-email-using-nodemailer-gmail-7cfa0712a799
  https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm
W3Schools. Node.js Send an Email. nd
  https://www.w3schools.com/nodejs/nodejs_email.asp