North American University

Project Report

COMP 3322 Software Engineering

Bulletin - NAU

Authors:
Albina Frolova
Mikhail M.Meskhi
Shohrat Muhamov
Bilal Onal
Bedir M. Tapkan

Supervisor:
Dr. Kemal Aydin

December 11, 2017

Contents

1 Introduction

Communication is a fundamental part of our lives. University life is full of busy people running around trying to get their agendas done. They depend on sending and receiving message instantly. Their main tool remains to be email. Unfortunately, email is not always reliable and a fast method of communication. Since the dawn of the age of the internet, messengers were created. Their goal was to facilitate communication by offering instant connection and message delivery. Our motivation for this project stems directly from the necessity of instant communication. Bulletin-NAU offers the capabilities of instant messaging connecting students to faculty members.

1.1 Requirements Analysis

1.1.1 Explain the need and how this project will address it?

Have you ever faced a problem and were not able to get a direct answer in the nick of time? Well we all have been there. Sometimes we face problems that we wish to resolve instantly, but reaching people capable of doing so may not be so easy. Bulletin-NAU will be able to connect students to faculty members at any time and anywhere.

1.1.2 Who are the end users?

This project is aimed at North American University students and faculty members.

1.1.3 Software Product Features

Bulletin for NAU has various features that enable people to communicate easily and effectively. Group chatting is one prominent feature. It automatically groups students, faculty, and staff (e.g. COMP 1411 students can join a automatically formed group called COMP 1411). On other feature is issue related grouping. Students or faculty members can create groups according to important events or issue on campus. Such as Student Services can create a group dedicated for resume workshops or even quick tutoring chats.

1.1.4 Software Performance, Portability, Security

Our application performance rates are quite fast and optimized. There have been no issues recorded on the iOS platform. Speaking of portability, we current are able to offer this application only for the iOS platform. We conducted a small survey and saw that most users have an iOS device. The survey yielded 59 percent iOS users versus 41 percent of Android users. Firebase 3 offers extensive security packages thus securing each users personal data well.

1.1.5 Database Requirements

As a means of keeping track of messaging and user data we utilized Google Firebase 3 as our database. Firebase gives you the tools to develop high-quality apps, grow your user base, and earn more money. We cover the essentials so you can monetize your business and focus on your users.

2 Project Organization

2.1 Software Process Model

We selected to follow the **Agile development** model for our iOS application. The Agile development process model favors adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change. The Agile manifesto consists of:

1. Individuals and Interactions over processes and tools
    (a) As university student ourselves we understand the product in and out thus giving us a complete advantage of understanding what needs to be implemented in order to achieve the goal of this project.
2. Working Software over comprehensive documentation
    (a) Our software was built with keeping in mind that importance of simplicity. The user must "naturally" understand the application and its utilization without having to explain every detail or action to take.

3. Customer Collaboration over contract negotiation

    (a) We consulted with the Computer Science Department faculty in order to specifically tailor the application to suite the needs of North American University student's needs and help the faculty member to be more helpful.

4. Responding to Change over following a plan

    (a) As we built the application, we tested each feature and made changes accordingly. It is very difficult to just write out a solid plan to follow. It is much more efficient to be able to respond to change during development.

2.2 Roles and Responsibilities

- Design
    1. Bedir M Tapkan
    2. Shohrat Muhamov
- Front-end
    1. Bedir M Tapkan
    2. Mikhail M.Meskhi
- Landing Page
    1. Albina Frolova
- Back-end
    1. Albina Frolova
    2. Bilal Onal
- iOS Development
    1. Albina Frolova
    2. Bilal Onal
    3. Bedir M Tapkan
    4. Mikhail M.Meskhi

3 Project Management Plan

3.1 Tasks

Each member of our team was assigned a task they are capable of doing plus a task they need to research in order to complete. This meant at the end of this project, each member would have left the team with more knowledge than they started with. The list above illustrates what member worked on which tasks assigned.

3.1.1 Design

The designed of our application stems from the three colors of North American University. Purple, white, and gray. The general layout of the application follows a simplistic idea. This simplicity helps the users understand the application faster and not get confused with the environment created by the application.

3.2 Front-end

The front-end of the application that assembles the design into interactive pieces of code was written completely in code rather in story time (Swift Development Interface of X-Code). This gave us the upper hand in fine tuning the interface and perfecting our design and applicability.

3.2.1 Landing Page

In order for a product to be marketed we decided to created a landing page that would advertise our application and explain it to customers. The application was written in HTML5 and CSS3. It is a good place to review the product and understand the functionality.

3.3 Back-end

For back-end we used Google's FireBase 3. Firebase 3 is a free database that can be easily implemented in mobile applications to facilitate data storage and user authentication. We create a database to store all the registrants and their private information. Also it helps the application in grouping same type of users into same chat rooms.

3.3.1 iOS Development

The main language used in the development process was Swift v.4.0.3. Swift is a very flexible language and modern power tool of mobile developers. The development process was facilitated with X-Code's iPhone simulator that helped us see amnd test the changes and new features we added immediately.

3.4 Timetable

The whole process took use about two months time. In September we started off with understanding the project idea and assigning tasks. The second half of September each member spent time understanding the APIs and languages they needed to use. In October we met and started building the initial skeleton of the application while still formulating the idea and adding/removing features. Towards November we had a alpha build ready to be tested. We managed to install the application on multiple phones and tested out its functionality. The database was working fine although we experienced some issues in user experience and design that we managed to fix up quickly.

4 Testing

4.1 Features Tested

We tested throughly every feature built in. The final product submitted contains 100 percent tested and working features. Some of the features that are worth mentioning are registration, login, chat grouping according to person type (student, faculty, staff, etc...), and file sending.

4.2 Features Not Tested

As mentioned before, there are no features that were not tested since this would cause us to ship a incomplete product.

4.3 Test Cases

1. Login: We tested the login feature by entering various emails and passwords to check whether they were recalled from the database correctly and logged in the user to his/her respective account. We had some

issues recovering login information after registration from the proper table but was fixed.
`Status: Passed`

2. Register: We tested the registration feature by testing what emails and passwords are applicable for the user to set. We then checked how the following was processed and stored in the database for each user under their unique user-id.
`Status: Passed`

3. Upload Profile Picture: We checked whether the application had permission to access the photo library and the photos were properly assigned to each user-id.
`Status: Passed`

4. Contact selection: This test checked whether we could see all registered users and were able to select them to start a conversation.
`Status: Passed`

5. Sending/Receiving: We tested sending and receiving text messages from and to different accounts. We also followed the database and made sure the messages were recorded in the correct table and had proper user-ids and timestamps.
`Status: Passed`

6. Logout: Tested logout function for users. The function worked as expected, closed the users session successfully.
`Status: Passed`

7. Grouping: We tried to automatically group users based on their class or topic. We failed to implement this feature due to the fact that we had to connect our application to the NAU Moodle database. Our application would request the table for a group of students and automatically create a chat room. (e.g. Application requests table of students in COMP 1411 and creates a chat room called COMP 1411).
`Status: Failed`

8. App Store: We have not uploaded the application to the Apple App Store for public download but have made the application available from the landing page.
`Status: Failed`

5 Additional Material

5.1 Screen shots

Screen shots can be found at this link: `https://goo.gl/1NvmNr`

5.2 Final Presentation

The final presentation video can be found at this link: `https://goo.gl/QRSQc4`

5.3 Code

The complete code for the application, landing page, and other can be found at this repository: `https://github.com/nau-bulletin/bulletin_NAU`