Moods – App for Champlain Students

By

Amir Osman

Étienne Bérubé

William Perron-Lafleur

Presented to

Mr. Amin Ranj Bar

420-204-RE, Integrative Project in Computer Science and Mathematics, section 00871

Wednesday, February 26th, 2017

Champlain College Saint-Lambert

**Table of Contents**

**Introduction**………………………………………………………………………………1

**Design**……………………………………………………………………………………..4

Analysis……………………………………………………………………………4

Algorithm………………………………………………………………………….4

UML Diagrams…………………………………………………………………………………..5

Graphical User Interface……………………………………………………………………………….…10

Object Design……………………………………………………………………………………11

Timeline………………………………………………………………………….12

Software………………………………………………………………………….13

**Features**……………………………………………………………..…………………. 14

**Conclusion**………………………………………………………………………………15

**Introduction**

As our final project we want to create a social network on android phones exclusively dedicated for Champlain students. Communication has always been something of paramount importance in the everyday life. During our college experience, we have noticed a lack of utilities to facilitate contacting the persons we care about. There is no way for students to get instant information about their college peers. Therefore, students who are looking to connect with other students are unable to do so to their liking. Considering students spend a lot of time at school, it is undoubted that such application would be extremely relevant and helpful in order for their life at school to be at its best state.

Additionally, we have also noticed that education is an unexploited domain in the programming world, even though education is essentially a domain everyone is involved at some point of their life. Hence, our goal is to make everyone’s experience at schools the best possible. We want people to have an optimal educative experience by providing them with a service that will allow them to facilitate their communication at school. Incidentally, what makes Moods special is that not only will it target the educative area, but also entertainment. Moods will offer students tools to help people study, but ultimately it will allow students to hang out together. Students wanting to eat for instance, will be able to connect with other students that are hungry. Students wanting to play outdoor soccer, will be able to organize an event. Students wanting to relax, will be able to reserve a room to lay back in. Moods has the opportunity to be successful because it provides simple solution to problems students meet and school. Our goal is not to provide something complex to students, as simplicity is complementary to efficacy. This is why each user is going to be able to reach out to other students within a click. We want students to be able to meet their friends and new people in the College more easily by using our application.

The lack of communication between students at school is essentially due to the fact that schools are too big. It is often difficult for people to know where their friends are, what they are doing, who they are with, what’s their schedule, etc. Therefore, the simple idea of meeting up with a friend is difficult, since you absolutely need to have their phone number and to have them use it exactly when you attempt to communicate with them. On the other hand, Moods would solve that problem by allowing students to publish their location and status/mood so that people don’t waste time anymore asking basic questions such as ‘Where are you?’ ‘What are you doing?’ to every single one of their friends. Instantly, the user will be able to see which students are available, what they are doing, and at what location they are (cafeteria, library, computer lab, etc.)

Analyze, define, and understand the problem

The major problem is that students' are having trouble connecting quickly with their friends within their college. The main reason why is that they have no way of consulting their peers schedule, thus it is extremely inconvenient as students don't know when they have common breaks with their friends. This prohibits the students from doing things they would prefer doing together, such as studying, eating, learning, relaxing, and the list goes on.

Our application would be the perfect solution to that problem as one of its feature involves storing every students' schedule in database. Using an efficient mathematical algorithm, our tool will be able to perfectly match which students have common breaks. In other words, our users will be able to know who's available when you are to do anything you want, easily. In order to display what is that thing you want, you will be provided a tool to update your mood, thus your friends will know what you are up to.

Additionally, colleges are important in size, thus locating your best friends can most of the times be time-consuming. Therefore, our application would remedy to that problem by providing an option to share your location with your friends, so that they know exactly where you are when they need your presence, in the matter of seconds.

Again, there is an obvious need to simplify the students' social life at college by helping them connect together easily and quickly.

The Mathematic Algorithm

*If you use a specific algorithm or formula (physics, math or chemistry), write the algorithm or formula* for *solving the problem, along with a complete description and find the Efficiency of the algorithm. (Keep in mind the report has to be complete for anyone with any background).*

*Explaining your design and the reason behind it. The explanation should include at least: UML (complete description of every class, every method, and relationship between them). You should include explanations similar to* [*http://docs.oracle.com/javase/8/docs/api/*](http://docs.oracle.com/javase/8/docs/api/)

*GUI (design of every single page in your project. For drawing your GUI, you can use any software). Design of every character or object in your project*

*TimeLine, including all steps from first week of classes until the day of delivery.*

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Plan Date | Assigned Person | Notes |
| Introduction | 19-Jan | Whole team | Team choosed |
| Discuss Project | 24-Jan | Whole team | Discuss the choice of project |
| Choose Project | 26-Jan | Whole team |  |
| Project Proposal | 31-Jan | Whole team | Project's features and used softwares |
| Install Softwares | 02-Feb | Whole team | Install and start getting familiar with Android Studio and GitHub |
| Familiarize with Softwares | 07-Feb | Whole team |  |
| Familiarize with Softwares | 09-Feb | Whole team |  |
| Design | 14-Feb | Whole team |  |
| UML diagrams | 16-Feb | Whole team |  |
| SQLite Database | 21-Feb | Whole team |  |
| Finish Proposal | 23-Feb | Whole team |  |
| Prepare presentation | Reading Week | Whole team |  |
| Oral Presentation | 07-Mar | Whole team | Summary of written report |
| Start canvas | 09-Mar | Etienne & Will | Design |
| Start database | 14-Mar |  | Database and Class implementation at the same time |
| Finish login page | 16-Mar |  |  |
|  | 21-Mar |  |  |
| Finish home page | 23-Mar |  |  |
| Finish Settings and profile | 28-Mar |  |  |
| Finish app canvas | 30-Mar |  |  |
| Finish database | 04-Apr |  |  |
| Start implement | 06-Apr |  |  |
|  | 11-Apr |  |  |
|  | 13-Apr |  |  |
|  | 18-Apr |  |  |
| Finish implementation | 20-Apr |  |  |
|  | 25-Apr |  |  |
| Finish networking | 27-Apr |  |  |
| Testing | 02-May |  |  |
| Testing | 04-Apr |  |  |
| Final Version | 09-Apr |  |  |
| Final Presentation | 11-Apr |  |  |

**Design**

Analyze, define and understand the problem.

Algorithm

Explaining your design and the reason behind it

UML diagrams

GUI (design of every single page)

Design of every object in your project

TimeLine, including all steps from first week of classes until the day of delivery

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Plan Date | Assigned Person | Notes |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Software used:**

**Platform:** Android Java

The application is expected to run on Android Devices. All devices using Android Lollipop and above will have an access to the application.

**Menu**: User profile

The user will interact with the application throughout multiple pages and menus. There will be a home page for the user to change his mood and status, yet the rest of the settings will be accessed through the settings page.

**Accounts**: Login Authentication

The users will be able to log from a database. A basic encryption scheme will be used to protect the user’s privacy. Using this database, the application will be able to retrieve the user’s information.

**Code sharing platform:** GitHub

The GitHub platform is used in order to efficiently work on this project. Using this service, one may share his codes and resources in real time with his coworkers.

**Code IDE:** Netbeans & Android Studio

**Netbeans:** This program is used to program the Database and the networking since the coding environment is better for this kind of programming. Android Studio is not made for such programming.

**Android Studio:** Android studio is used in order to manage and program everything that is in relationship with the activities. This program is optimized for a visual environment in order to code XML and Java files efficiently.

**Features**

* Location of users (sends to selected contacts)
  + Uses GPS location from device (API integrated in Android Studio )
* Mood & Statuses (Looking for someone to Eat, Study, Tutor, Lift/Go Home, Chill)
  + Helps you communicate with friends
  + Stored in database (resets every 24h)
* Map of Champlain
  + Drawn by us (JPG format)
    - Ask school for plans
  + May use GoogleMap SDK
* Notifications to cellphones
  + Send if the friends is near or accepted a “poke request”
* Access to vibrations or other phone components
  + Access to GPS
  + Access to Vibrations
* Networking
  + Use SQL databases (SQLite implemented in Android Studio)
  + Store user information on server
    - User name
    - Email
    - Pictures
    - Schedule
  + Store user information on phone
    - ID
    - Temporary mood

**Conclusion**