**SEG 3904 Project Proposal**

**Project Title: *Symatree* – Mobile based tutoring application for students in university of Ottawa**

Name: Pramity Roy

Student Number: 7382156

E-mail: [proy061@uottawa.ca](mailto:proy061@uottawa.ca)

Professor: Miguel Garzon ([mgarz042@uottawa.ca)](mailto:mgarz042@uottawa.ca))

**Overview:**

The main purpose for this project is to develop a mobile application for tutoring services within the University of Ottawa. Students enrolled in an university often finds difficulty in many courses and are often discouraged to seek help due to limited resources available within the university. Mainly, every student has at least once experienced their professor and/or teaching assistant to lack quality in teaching. Low quality teaching results poor gain in knowledge and thus outcomes many students to either fail or do poorly in a course regardless of the ease of the material. Therefore, it raises our attention to find a solution and hence make an application that allows students to get access to tutors that have excelled in the particular course(s) they are currently struggling with.

*Tut*, will allow students to either take advantage of courses they have excelled in and earn money electronically by becoming a tutor, or pay tutors by learning and getting help from them regarding course materials. This mobile application hence varies for the two types of users: tutors and students. In addition, Tut allows the opportunity for a user to be both tutor and student.

Tut however makes a restriction for becoming a tutor for security reasons and therefore is controlled by the system administrator. The system admin must assess and access the required documents such as transcripts and police records, and decide whether an individual is qualified to be a tutor. Upon meeting the minimum requirements, an individual can be registered to tutor a particular course. On the other hand, upon creating an account, a user entitled as a student, can navigate and browse through the application at free cost and check for available tutors for a specific course. If available, students can then either sign-up for one to one session online or in-person meeting or group sessions that could be either on-line or in-person. In addition, payment and money is transferred only after when a meeting has been done successfully between a tutor and a student.

There already exists a variety of tutorial applications but however courses vary from universities based on the context. Therefore, it is the only application existing where the context is controlled since both the student and the tutor have learned the same materials or even had the same professor. Also, most application does not provide the opportunity for students to be able to request a tutor right away at the place of their desired location (e.g their house) and with this application, it is very much possible to do just as how Uber eats work. This application will therefore be the most efficient tutoring application existing since it connects both the student and the tutor effectively, making learning more enjoyable and resulting excellent grades.

**Learning Outcomes:**

At the end of this project, the student will have learned to:

* Program in a different programming language than those that they have been exposed to such as Java
* Apply a wide variety of design patterns, frameworks and architectures in the design of mobile applications such as MVC pattern.
* Discover and use existing APIs, such as Google Maps for location services
* Handle and arrange software engineering projects, including project scheduling and planning.
* Individually Work on software engineering activities in order to accomplish particular tasks.

**Technologies:**

To meet the demands and to increase user experience satisfaction and ease, a mobile application is considered to be highly suitable for this project rather than a web application. Justification of choice is due to:

* Availability: Students can be available online even when they are not in class. This allows students to be flexible and arrange meetings quicker with tutors.
* Accessibility: Students can access the application through mobile devices easily in class and ask questions instantly to tutors for clarification.
* Ease of meetings: By having a mobile application, users can be able to see by using google map, where the meeting location is. In addition, a user can see via google map constantly how far away the other user is from the meeting spot in case one user arrives late. Upon certain time waiting, a meeting can be cancelled.

Hence, the technologies to be used to develop the mobile application are:

|  |  |
| --- | --- |
| Ionic Framework | Free, open source mobile SDK for developing web applications |
| CSS | a style sheet language used for describing the style of a document written in a markup language. . Ionic supports CSS |
| HTML | a standard markup language to create web applications. Ionic supports HTML |
| JavaScript | Popular web-language for its ability to do dynamic tasks within a static web page by making it more interactive. Mainly useful for handling requests and performing actions without loading multiple web-pages. Ionic supports JavaScript. |
| Font Awesome API | It is an iconic font and CSS framework. It gives scalable vector icons for which can be easily be customized. |
| Google Fonts API | It is an open-source API that gives most appropriate fonts to use on web sites. This permanently makes sure a consistent view cross browsers independent of the availability of the font on the user device. |
| Mongo Database | It is a back-end development tool. It is mainly used for database. |

**Resources:**

* Flanagan, D. (2011). Javascript: the definitive guide. O'Reilly media.
* DuCkeTT, J. (n.d.). HTML & CSS Design and Build Websites. JoHn WiLey & SonS, inC.
* Technology websites mentioned above.

**Deliverables**

|  |  |
| --- | --- |
| **Deliverable** | **Weight** |
| Initial application description and requirements. | 10% |
| Prototyping | 5% |
| Phase-1 of the software. Admin View of the software | 25% |
| Phase-2 of the software. Tutor View of the software | 25% |
| Phase-3 of the software. Student View of the Software | 25% |
| Proper use of configuration management and tracking tools. | 5% |
| wiki with final requirements, architecture, challenges/resolutions, code structure overview, self-assessment of learning, and future work items. | 5% |

**Work Plan (135 hours)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Meet?** | **Action** | **Hours** |
| 1 |  | Project plan, expectations, and technologies. Environment set up and mockup. Application description and requirements. | 10 |
| 2 |  | Hello world system | 15 |
| 3 |  | Initial version, with use of technologies. | 10 |
| 4 |  | Architecture development. | 15 |
| 5 |  | Implementation of selected set of features/scenarios. (Phase-1) | 15 |
| 6 |  | Implementation of selected set of features/scenarios.  (Phase-2) | 15 |
| 7 |  | Implementation of selected set of features/scenarios. (Phase-3). | 15 |
| 8 |  | Implementation of remaining features/scenarios. | 15 |
| 9 |  | Implementation of remaining features/scenarios. | 10 |
| 10 |  | Report drafting and software inspection. | 10 |
| 11 |  | final software delivered. | 5 |