### **CSCI 4176 & CSCI 5708**

### **Mobile Computing**

# **Project Update Form**

# Due: In Lab on Wednesday October 31st, 2018 **NUMBER 5**

1. **Participation (Grey Columns Filled in by TA):**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Name (Print)  **Adjust** | | Banner Number | OK? | Git  Data | Comments |
| 1. |  | Jigar Joshi | B00812722 |  |  |  |
| 2. |  | Shivani Desai | B00799169 |  |  |  |
| 3. |  | Bola Okesanjo | B00786268 |  |  |  |
| 4. |  | Vivek Shah | B00799155 |  |  |  |
| 5. |  | Ryan Stevens | B00695460 |  |  |  |

|  |
| --- |
| **Mark / 5** |

1. **Progress Report**

List completed work; identify partially completed work by %.

The project is currently in the first phase of development, where all the required development software has been  installed on a developers’ local machines.

The team will make sure that all the systems are up and running to ensure smooth out development in upcoming weeks. The next phase of the project is to start implementing basic functionality.

The team had several healthy discussions on what controls should be given to users, the requirement of login page, how the pages should be navigated through out the application, and design and implementation of the user interface. The UI designing was done using UI Mockup software *Axur RP.* The team has also made project and uploaded it to GitHub to ease out consolidation of each team member’s contribution. UI Implementation of Detect media, add media and search media has been started. Database was created to generate APIs in order to do CRUD operation.

**Completed Work(20%):**

1. Installed Ubuntu OS on virtual machine
2. UI designing using Axur RP
3. Database created
4. UI Implementation of Add media, Detect Media and search media

**Individual Contribution**

Jigar has installed prerequisite software on his local machine to create APIs which can be used to do CRUD operations.  Part of this has been configuring and installing a Linux on a virtual machine. He has also created a database design diagram which is useful to identify entity relationship between tables.

Ryan’s main contribution to the *BooYah!* project has been to use the UI Mockup software *Axur RP* to create the wireframes seen in Appendix A. Wireframe pages were mostly drafted by Ryan, and then refined during group meetings, with Ryan being responsible for making edits quickly. Otherwise, Ryan has contributed to the group by being responsible for booking meeting rooms.

Vivek and Shivani’s has analyzed the wireframes created and started implementing user interface based on it. Till now, they have successfully converted three wireframes into actual pages.

Bola has worked on storing media files, divided media into many parts to use voice recognition feature.

He and Jigar would start working on creating APIs in next phase.

|  |
| --- |
| **Mark / 5** |

1. **Identification of Problems**

List any problems and status; e.g., solved, not solved.

### Whether or not to allow user-added media entries

During discussion about how the user would perform ratings, a significant problem was identified: how users would keep their *BooYah!* timer reliability synced with media content if they used a source with a different duration; for example, user A creates an entry for “The Simpsons, Episode 12”, based on the duration found on a DVD version, but user B tried to rate that same episode while watching a version on Netflix, there a short logo is displayed before the episode begins. It was determined that the “perfect world” solution to this problem would be to rely on the “sound recognition to identify time indexes” bonus feature, which would require that content entries be added by the developers.

A healthy debate was then sparked over whether or not to accommodate “non-indexed” content, and whether or not to allow users to add their own media entries. Beyond user-added entries not working with the eventual sound recognition feature, it was identified that such media entries could prove problematic if users they entered “wrong” durations or, still, had different sources on the media. On the other hand, not letting users create their own entries would restrict what media content they could rate with the app. A few solutions were considered, including allowing for different entries for the same media from different sources; this solution was discarded, however, since it would lead to a bad user experience if a user needed to find the “correct” entry from a long list.

The solution agreed upon was to allow only one “authoritative” entry of a given piece of media content, with a duration that is taken to be correct when the first user submits it. To then account for this duration being incorrect, a “report” button would then be provided, to prompt the *Booyah!* app developers to identify the “correct” duration for the media.

### *BooYah!* rating screen design is not final

As of now, the design for the *Booyah!* screen is not final. This is an important part of the app, and will require a great deal of prototyping to get right. Thus, the final design of this part is being deferred until we have a working prototype that we can feel and experiment with.

|  |
| --- |
| **Mark / 5** |

1. **Request for Changes in Functionality**

Be specific in identifying what you can complete.

In the first phase of application development, we have successfully managed to create UI designing. As we mentioned in project proposal, there is no change in base functionality. We would be focusing to finish the base functionalities in the next phase of application development.

As per Tami’s suggestion, we have removed some undoable( Due to time constraint) and irrelevant features from expected functionality.

Below are the list of base and expected functionality we would deliver at the end of this project.

Base Functionality

* Start, pause, and adjust time on a timer
* Create new ratings for media content
* Search existing media content entries
* View individual content profile and graph of user’s rating

## Expected Functionality

* View audience’s aggregate ratings for individual media content
* Generate other graphs in a report of the media (ex. pie chart)
* Share graphs and opinions of select moments to social media
* Delete a *BooYah!* rating the user has created
* Use voice commands to control timer
* View “Terms of Service” and “Privacy Policy”

|  |
| --- |
| **Total: Out of 15** |

TA Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Instructor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_