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

### **Mobile Computing**

# **Project Update Form**

# Due: In Lab on Wednesday November 21st, 2018 **NUMBER 7**

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Name (Print)  **Adjust** | | Banner Number | OK? | Git  Data | Comments |
| 1. |  | Jessica Castelino | B00804805 |  |  |  |
| 2. |  | Jismy Johnson | B00813344 |  |  |  |
| 3. |  | Navneet Singh | B00810744 |  |  |  |
| 4. |  | Sharon Alva | B00813350 |  |  |  |
| 5. |  | Shubham Narang | B00810159 |  |  |  |
| 6. |  | Utsav Soni | B00812689 |  |  |  |

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

1. **Progress Report**

**Overview:**

To Do List is an application which helps users to maintain and monitor their day-to-day tasks. Our target audience for this application are the students currently studying in Halifax. The main functionalities of To Do application include: add a new task, update an existing task, delete a task, tag a location to a task, and set a reminder to each task. Other functionalities include the use of camera feature for quickly adding a task and getting task notifications.

**Status:**

The application has the **create, update** and **delete** functionalities implemented. A new task can be created with title and subtasks. The home screen of the application displays the created lists of tasks. Implementation of the list view of tasks is completed. Tasks are displayed in a list view in the order they are created. Tasks details are displayed by fetching the data from the database. It is dynamically displayed by making use of Live Data that binds to the data and updates the displayed list.

Task creation is implemented on click of a floating button, where a pop-up appears. It then prompts for the task name. The pop-up window also adds any subtasks that need to be added under the main task. A new pop-up appears to add subtasks. At a time, only one subtask can be added to the task. Once the task is completed, it can be checked off from the list. The delete option is implemented to remove individual subtasks from the main task.

A navigation drawer is implemented to provide quick access of all the pending and completed tasks. Currently, a static menu drawer is implemented which has options; pending tasks completed tasks, and feedback. However, implementation of these options is pending.

Every task created can be tagged to a specific location such that the user gets notifications when near that location. For this feature, the create pop-up has an option implemented to tag a location. On click of this option, it redirects to a new screen where you can search for a location. This was implemented using **Google API**. For this, the application requests for Internet permission from the user. When searching for a location, a set of places are displayed in a drop-down list. This functionality is integrated and stored in the database.

Another feature implemented within the task creation is adding a reminder date. This option is placed next to the tag location option. A calendar opens where the date can be selected and saved against the task.

The camera feature is implemented for capturing images of notes, checklist and shopping list. It provides the user with an alternative to add an image directly rather than text. The user can also add an already taken image from the device’s gallery. The image is added as a content to the task.

The current location of the user is taken by using the GPS feature of the device. This is implemented using **FusedLocationProviderClient API** of Google play services. For every task that is tagged to a location will display a notification when the user is near the tagged location. A broadcast notification is implemented where using the current coordinates and the location coordinates are matched to display the notification. This feature is implemented and integrated with the application. It fetches the coordinate values from the database to compare it with the user’s current location. User’s current location coordinates are to be collected for displaying the location-based notification. For this, the application runs in the background to collect the updated location. This feature has been implemented as a separate functionality, though it is yet to be integrated.

As the team was concentrating more on implementation rather than the user interface, the interface isn’t implemented as per the wireframes. The team will now focus on the applications’ user interface in the coming days.

All the individual modules that are implemented till date have been unit tested. The test cases have been included in the GitLab repository.

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

1. **Identification of Problems**
2. Internal project updates:

During the initial development phase, there was a lot of confusion on the amount of work completed. Some team members sent updates by e-mail, others on WhatsApp and there was no single place where we could keep a record of what has been done and what is coming next. To resolve this issue, we made use of the Slack app. All the communication regarding project updates were restricted to Slack to ensure that everyone was on the same page.

1. Adhering to coding conventions:

After the initial commits on GitLab, it was noticed that the code developed violated the Java coding conventions on various levels. As a team, we investigated the problem, invested our time to read the Java coding conventions defined on the Oracle website, and ensured that all rectify it without affecting the timelines.

1. Integration efforts:

Integration is crucial to ensure that all parts of the project fit together. Our milestones incorporated the development of individual tasks, however, the integration efforts required were not considered. This affected our timelines.

1. Bugs:

Minor bugs in the code were detected during the implementation. This was then resolved by discussing within the group. Using this method helped in accomplishing the deliverables faster, as a different approach for a problem, solved the issue.

1. Differences in Gradle version:

Some team members found a difficulty to run the code pulled from GitLab after the initial commits were made. On investigating the problem, we realized that there was a mismatch in the Gradle versions used by every team member. As a result, the code updated to GitLab could not be executed and problems related to Gradle sync were encountered. We updated the Gradle to the same version on our machines to resolve this issue.

1. Change in design:

Initial wireframe was difficult to implement and due to our limited android knowledge, we couldn’t replicate the design completely. This problem was resolved by understanding the Android design standards and re-implementing the design.

1. Database restructuring:

The database was normalized till NF form. We had kept separate tables to store the individual tasks and the reminder details. The location-based reminder functionality, which runs as a background service, compares the user’s location with the location details stored in the database. This required a constant join operation which was inefficient and unreasonable. Hence, database restructuring was done.

1. Reducing memory footprint:

The location-based reminder functionality runs as a background service, constantly polling for the user’s GPS coordinates and comparing it with the destination location stored in the database. During the analysis phase, we realized that such a service can increase the memory footprint.

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

1. **Application Features**
2. Add new task:

The application allows users to add a new task such as “Grocery”. This task can have multiple sub-tasks such as “Milk, bread, cheese” added under Grocery. The task title and its sub-task elements are stored in the database.

1. View tasks on home screen:

Every task created will be displayed on the home screen of the application. The order of display is as per the creation time. Instead of displaying all the subtasks at home screen, only the task title is displayed. This removes clutter on the home screen.

1. Checking / unchecking task:

The application allows the user to check off the task is complete. Once the task is complete, a user may want to either check or uncheck the task. This gives the user an idea of how many tasks are completed. It helps the user to focus on other pending activities.

1. Update existing task:

The application allows the user to update an existing task. As user freedom and control is of high importance, users can update an already created task. The app allows editing the task title, location, reminder date and the subtasks listed. This helps users in updating and changing as per their needs.

1. Delete task:

The application allows the user to delete an existing task. Once the task is selected, an option to delete is available on the menu bar and user can delete the task. There are times when the task is of no value, too old to maintain, or an incorrect task detail. This feature allows users to delete unwanted tasks.

1. Camera feature:

The application allows a user to upload To Do list as an image. It provides two options to get an image. First, a user can use his camera to click a picture of a list and the application will upload that in the database. Second, a user can upload an existing picture from gallery.

1. Location tagging:

Location can be tagged against a task. User searches for the address or location and a list appear which is taken from Google API. Once the user selects the task, details are stored and reflected against the task that location.

1. Location-based reminder:

The application gives a provision to the user to set a location for every task. In the future, if the user happens to be around that location, a reminder will be sent to the user in the form of a notification.

1. Support & Feedback:

A simple feedback form will be displayed for the user to send their concerns and queries regarding the application. This feature will be accessible from the drawable menu. By providing support regarding the app, it helps not only them with their concerns, but also us as to how effectively the application was developed.

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

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

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