

# Team Project

## ICT2105 Mobile Application Development Spring 2020

### Objectives

The project is a trimester long implementation of a real, fully featured mobile application in Android. The main objectives are to:

- Learn to develop a working mobile application in Android
- Gain experience with some Agile software development methods such as Continuous Integration (CI), stand-up meetings and sprints
- Work in a team to develop a mobile application of moderate complexity

### Requirements

The mobile application should be a 100% native Android application. No hybrid applications or hybrid app frameworks should be used. The form factor can be a phone, tablet or wearable (watch), or any combination of form factors.

The design and features of the application should correspond to the given design theme. On top of a UI implementation, the app should aim to employ **three or more** of the following mobile features as suitable or relevant: database with multiple tables or other suitable content provider, sensors (e.g. accelerometer, GPS, NFC) multimedia, camera, animations, networking, multi-threading, or other advanced implementation of features such as VR/AR, computer vision (CV) or machine learning (ML).

Proper software engineering practices should be followed. This means version control with Git in Github with regular commits. The code should be well commented and self-documenting, and the code must compile and build at any stage. CI should be running on the repository. Unit and integration tests should be written and preferably be running on CI.

Documents and any non-code items should be committed in the documents repo. Third party libraries, code or tools used in the application should be clearly identified, commented in the code and listed in the final report or document.

The entire project will be graded based on the grading criteria at the end of the trimester; however, zero will be given for that grading criteria section if there are no deliverables at the end of the corresponding phase.

## Deliverables & Dates

Week	Date	Milestones/ To Do	Deliverables	Details
3	24/01/2020 2359hrs	<u>Project Proposal</u>	Initial project proposal User Stories	1-page proposal and ~2-3 pages for user stories (~4-5 pages total)
5	08/02/2020 2359hrs	<u>Design Phase</u> To Do: Design UI, Decide product and sprint 1, 2, 3 backlog	UI Prototypes Product backlog (aka feature list) Sprint backlogs Initial software architecture	Updated proposal and user stories with acceptance criteria Product backlog (from user stories), Sprint 1, 2, 3 backlogs (from product backlog) Software architecture diagram (~10-15 pages total)
7	22/02/2020 2359hrs	<u>First Sprint Completion</u> To Do: Sprint review, reassign sprint backlog	Revised product backlog Revised sprint backlogs Revised diagrams	Updated product backlog Updated diagrams Updated Sprint 2 & 3 backlogs
9	07/03/2020 2359hrs	<u>Second Sprint Completion</u> To Do: Sprint review, reassign sprint backlog	Revised product backlog Revised sprint backlogs Revised diagrams	Updated product backlog Updated diagrams Updated Sprint 3 backlogs
11	21/03/2020 2359hrs	<u>Third Sprint Completion</u> To Do: Sprint review, reassign sprint backlog	Revised product backlog Revised sprint backlogs Revised diagrams	Updated product backlog Final sprint backlogs Updated diagrams
13	30/03/2020 0900hrs	<u>Final Product Release</u>	SW & Code (in repo) Report & Slides Product Video	Report (see report checklist) Presentation slides Product Video (Post on fb page)

## Definitions

**User story** – a description from the perspective of the end user of the system that captures what a user does or needs to do with the system to achieve a task. User stories are used to define the functions of the system and determine requirements in a simple concise way.

**Product backlog** - Requirements that a team maintains for the given product. It consists of features, bug fixes, non-functional requirements, etc.—whatever must be done to successfully deliver a viable product. The requirements and features are normally obtained from the user stories.

**Sprint backlog** – List of tasks identified by the team to be completed during the 2-week sprint. This is a selected subset of product backlog items in the form of user stories decomposed into smaller tasks.

**Software architecture diagram and design**– One or more diagrams describing the components of the application, design justifications and decisions made, and any accompanying class or UML diagrams. Use any knowledge from ICT2106 Software Design for this.

## Plagiarism

C'mon, you guys don't need to be reminded that it's against University policy to copy code, assignments, quizzes and answers from others, the Internet and anywhere else, or sub the work out to another contractor. Any copied work between people/teams will be graded and the grade divided by the number of people who "shared" the work. If copied from the Internet or external sources, or it is discovered that the work does not belong to the owner, the project receives zero. **Just don't do it.**