ICT2105 MOBILE APPLICATION PROGRAMMING
TEAM PROJECT GRADING CRITERIA (40%)

	Excellent (A) (8-10 marks)	**Good (B) (6-7 marks)**	**Average (C, D) (3-5 marks)**	**Fails to Meet Expectations (E, F) (0-2 marks)**
Presentation, Demo & Report (10%)	Presentation is clear and logical. Excellent coverage and good explanation of design, engineering process, features and methods. Appropriate, interesting and complete demo of use cases and features.	Presentation is generally clear, mostly good report with design explanation and details, some minor errors and ambiguities or omissions. A few minor points may be confusing. Good demo of features.	Presentation is unclear, with less structure. Report organization not well thought out. Partial explanation of design, process and methods, some errors and ambiguities. Very little demo.	Presentation is very confusing and unclear. Brief and minimal report with little explanation and details. Nothing much done.
SW Engineering Practice (10%)	Regular commits to main branch, code clearly commented, conforms to coding standards, appropriate method names and class structure, build is not broken. All unit and integration tests written and running.	Occasional commits, code sometimes commented, generally conforms to coding standards, appropriate method names and class structure, build sometimes broken. Some tests written and running.	Irregular commits, partial comments, sometimes inappropriate method names and structures, build broken most of the time. Some or no tests written, may not run.	Irregular commits, partial or no comments, inappropriate method names, code does not compile. Build broken. No tests written.
Technical & Implementation (10%)	Significant implementation to support features/use cases, appropriate choice of toolkits/API, algorithms and or complete implementation of modules. Good software architecture design and component choice.	Good implementation to support features/use cases, appropriate choice of toolkits/API. Good software architecture design and component choice.	Minor implementation for the sake of convenience, minimum testing, may not work. Poor internal software architecture design.	Total copy of demo code, or minimal coding done. Poor software architecture.
Design (10%)	Out of the box creative (novel) design, good use cases, or interesting perspectives. Interactive and usable UI design adhering to HCI principles, with few or no issues. The video demos the features in an entertaining and useful way.	Interesting, interactive, appropriate and useful designs and features or mashups. Good UI design adhering to most principles. Possibly some usability or design issues. Video demos the features in a useful way.	Very few features and straightforward boring designs. May not conform to mobile screen or have some usability issues. Video does not adequately demo features.	Total copy of design idea, or nothing much done. No video or very short video.

PEER & INDIVIDUAL ASSESSMENT: Delta Multiplier

- Students evaluate peers within the group based on teamwork and contribution
- Instructor acting as engineering manager evaluates each individual based on contribution to the project and completed features

Presentation, Demo & Report (10%)

- Well-paced and structured presentation
- Structured and well written, readable report
- Complete actual app demo of use cases and features
- Good explanation of design, features, software engineering process, software architecture within the report

Software Engineering Practice (10%)

- Regular code commits from all team members throughout the trimester with meaningful commit messages
- Well commented and self-documenting code
- Code conforms to Java coding conventions listed at: https://google.github.io/styleguide/javaguide.html
- Code compiles and builds (build should not be broken)
- Continuous Integration (CI) should be running on the repository
- Unit and integrated tests are written for majority of features
- Tests run automatically on push to repo

Technical & Implementation (10%)

- Well tested; No obvious bugs
- Modular and well-structured code
- Good internal software architecture design
- Complete and significant implementation to support features or use cases
- Appropriate choice of toolkits, API or external libraries (To be listed within report)
- App employs **three or more** advanced mobile features (networking, database, sensors e.g. camera, gps, etc.) appropriate to the use cases
- (BONUS) Complex implemented component with special algorithms, or other advanced features e.g. ML, CV, VR, AR (mostly using a library or copying code from somewhere doesn't count). This component should be clearly documented within the report.

Design (10%)

- Creative and novel app design with interesting perspective
- Interactive and usable mobile app user interface without design issues or flaws
- Video is watchable and entertaining, and demos the main app features

Peer & Individual Assessment: Delta Multiplier

- Students evaluate peers within the group based on teamwork and contribution
- 100 points assigned per team member within the team, represents 100%
- Instructor acting as engineering manager evaluates each individual based on contribution to the project and completed features
- E.g. 4 members in a team, you get 300 points to evaluate the other 3 team members (apart from yourself)
- Assuming equal contribution, each team member gets 100
- Peer delta is the average of the received score from your teammates divided by 100
- Instructor delta is the assigned score of the instructor based on evaluation of the individual's contribution to the project
- Project Score Formula = Team Score * Peer Delta * Instructor Delta