North South University
Department of Electrical and Computer Engineering
CSE 327 - Software Engineering
Project Evaluation
Dr. Nabeel Mohammed (NbM)

General Details:

1. Each group has to work on a GIT repository on Bitbucket (https://www.bitbucket.org/).
    a. Each and every student is expected to have a Bitbucket account by the end of the first day.
2. Students are expected to form groups by the end of the first week. The group details should be emailed to me with the subject “CSE327 Section {X} Group Details” ({X} is the section number) ***before the second class of the semester***. In the email body, for each group member mention their Full name, NSU email address, Student ID.
3. Each group will create a git repository on bitbucket and they must add me as an admin to that repository. Use nabeel.mohammed@northsouth.edu to add me.

Work Practice:

1. Each student is expected to work on the project consistently, at least 5 days per week for every week in the semester. Each day’s work should be pushed to the repository by the end of the day. **This is the minimum expectation,** and failure to do so will result in a zero mark in the project.
2. When deciding the contribution of a group member, I will simply only use the GIT records as truth. Your verbal statements will be counted only if they match the GIT commit records.
3. One person committing/pushing someone else's work is not allowed under any circumstances. You must fix your GIT-related problems/challenges. No excuses will be accepted at this year-level.
4. Groups can attach a Trello board to the Bitbucket repositories, or use some other form of project management tool. The board should contain the project plan outlining the list of tasks, their status, and responsible group members. Marks will be severely deducted if this is not done.

Project topic-related initial information:

1. I will select and describe the project topic. Each group will work on the same topic.
2. The project topic will require
    a. A well-designed data model
    b. Integration with external libraries/systems/API
    c. Integration with frontier technologies
    d. Exposing system functionality via API
    e. Mobile application that will use the API exposed from your back-end

    f. OAuth
3. The number of features will be small but complex.
4. Automatic unit testing. Start learning how to do this from day one.
5. Do not expect things to “just work”, they hardly ever do. I expect to see GRIT from each and every student. It is not an academic prerequisite but a character prerequisite.

Technical Requirements

1. Each group **must select** and **properly use** a back-end framework. There will be no marks given if this is not done. Each and every group member is expected to learn (if necessary) and implement the back-end making full utilization of the framework.
2. A mobile application will be required. Multi-platform solutions are not allowed. It must be an application not just a web view.
3. The mobile application and the web must together be a unified product. Showing them piecemeal results in severe deduction of marks for all members.
4. You are free to use a non-relational DB if you can justify that your data is not relational.
5. Good and modern UI design
6. Support for multiple target platforms
7. Code and Comment Quality

Submission and Evaluation Items

1. Software Requirements Specification (SRS) - Hurdle.
2. Software Design Specification (SDS) - Hurdle
3. Demonstration 1
4. Demonstration 2
5. Final Demonstration
6. Installation Instructions and Help Manual

Note 1 - SRS, SDS, Code, Installation instructions, Help Manual and any other submission items will be accepted through the GIT repository)

Note 2 - Group members are not necessarily going to get equal marks.

Group dynamics

1. Any group member-related complaints must be lodged by emailing me the details and related proofs/documents that effort was made to communicate with the member.
2. Complains made during the demonstrations will result in a deduction of marks for every group member. Problems do not manifest at the time of the demonstration. Complaints should be made at the earliest possible time.
3. The project is small enough so that one person working regularly should be able to finish it completely. When judging the project output of a group where one or members did not cooperate, my judgment will be made on what is it that the remaining members could have done if they themselves worked regularly. Marks will be given to members who worked (claims backed by GIT records).

A bit more information

SL#	Item	Notes
1	Submission and completeness of SRS **(Hurdle)**	● Submission via GIT repository ● Must be in HTML format ● HTML should be printable, in a good printable format, via the browser. PDF downloads are not acceptable ● Must contain all relevant use case diagrams, expanded use cases, sections, screen mock-ups ● Template available in Google Classroom ● This is a hurdle. No further marks if this is not good.
2	Submission and completeness of SDS **(Hurdle)**	● Submission via GIT repository ● Must be in HTML format ● HTML should be printable, in a good printable format, via the browser. PDF downloads are not acceptable ● This is a hurdle. No further marks if this is not good. ● Required Sections are given in Google Classroom
3	Consistent and proper use of the GIT repository (for documents and source code)	● There are no marks for “just using git” ● To get any marks here, you must use the git repo properly and consistently. I expect students to work on the project 5 days per week, every week of the semester. ● Each student should be pushing their daily work to the repository every day.
4	Proper utilisation of a popular server-side software framework	● **No framework, no good grade.** ● Trying to fool the faculty member with a superficial use of a framework will result in negative marks. ● Start learning from day 0. Consider this as an instruction to start learning. Each member will be expected to have mastery.
5	Interesting feature	Interesting to me. Will be given when project topic is discussed
6	Use of a Federated Identity provider	● Just one is sufficient, e.g. just “sign in with Google” will do.

7	Source code comments (coverage and quality)	● Comments should say why a piece of code exists, not what it does. ● Header comments for each class, each method is required.
8	Unit tests (coverage and quality)	● Start learning how to do unit testing from day 0.
9	UI Design (ease and beauty)	● Better be the best UI I have ever seen.
10	Support for multiple target platforms	● The product must be available to use from multiple platforms, i.e. web and Android, Android and iPhone, Web and Watch etc. ● You must use native frameworks. Multi-platform frameworks are not acceptable.
11	Project Management	● Highly recommend you all use Trello regularly. I will view your Trello board and activities for this assessment.