Project Specification Document

As part of the CS 102 course, you need to create a project. This document specifies the technicalities for that task. The project is composed of several stages:

1. Group/Project Selection Stage
2. Requirements Stage
3. UI Stage
4. Detailed Design stage
5. Implementation Stage

Group/Project Selection Stage

You have to create a group of 4 or 5 people. You will decide on your project topic. The possible projects list is provided on the Moodle page. You could come up with your own project ideas as well. You have to prepare your project using the **Java** programming language and have to follow the **MVC (Model-View-Controller)** architecture in your projects. You are extremely encouraged to use an **IDE** (i.e. IntelliJ IDEA) during implementation to accelerate the process. You are advised to **independently learn** new tools and frameworks and use them in your project. To name a few, you may learn and use SQL (i.e. MySQL, PostgreSQL), NoSQL (i.e. Firebase, MongoDB) database systems; Cloud Storage systems, or Client-Server architecture for Multiplayer Games in your projects. You will have to use **Git** and **Github** in your project (see Github header).

Requirements Stage

You have to come up with the requirements for your software. That is, you should list all the features that you plan your software will support. Provide as much detail as possible.
Specifically, you should:

- use the provided template and fill it completely
- organize your report neatly (incl. spaces, margins, etc.)
- create grammatically and syntactically correct report
- properly cite the external sources and follow a citation style consistently (e.g. MLA, APA)
- give a brief description of the project in the introduction
- discuss similar works and the features they have
- talk about your software's features in detail
- discuss your targetted users
- mention the advantages and benefits of your system
- compare your software with existing ones

UI Stage

You should create a mock-up of your user interface. It should reflect what your application will eventually look like and cover all of your requirements.
Specifically, you should:

- use the provided template and fill it completely

- organize your report neatly (incl. spaces, margins, etc.)
- create grammatically and syntactically correct report
- properly cite the external sources and follow a citation style consistently (e.g. MLA, APA)
- create a UI design that completely covers your requirements and gives an excellent idea of the final product
- create a UI design that one can figure out with almost no effort (easy-to-use)
- create a UI design that makes use of a minimal number of UI elements without sacrificing clarity (minimalist)
- create a Ui design that requires as few interactions as possible (efficient)
- provide a graphical overview (i.e. sitemap) to demonstrate the dynamical behaviour of your UI
- discuss the concerns regarding your UI design
- show us you know how a good design should look like

Detailed Design Stage

You should create a detailed design document which is the last document before the implementation. Your documentation should ideally help you implement the project easily. You are going to talk about the classes, methods, and attributes of your project as well as UML diagrams visualizing all these. You should at least create a Class Diagram. Any other diagram which will help us understand your design is welcome. Avoid inserting your UI (User Interface) classes into your UML diagrams and do not mention it in your report.
Specifically, you should:

- use the provided template and fill it completely
- organize your report neatly (incl. spaces, margins, etc.)
- create grammatically and syntactically correct report
- provide a good amount of classes, attributes, and methods which can be used to implement a major portion of your project
- follow the MVC pattern
- create classes that have well-defined single responsibilities each
- create a UML Class diagram with classes, attributes, methods, and relations (e.g. inheritance, aggregation, etc.) properly defined.
- should explain your classes in detail (give details about methods and properties that are significant to your project, you do not need to talk about getters/setters, etc.)
- divide the labor and share the classes among the group members (discuss it in the report)
- discuss concerns regarding the detailed design of your project

Implementation Stage

With all the necessary documentation prepared, you can now start implementing your software. Divide the workload evenly so that everybody in your group has a task to do. You should use Github. Refer to the Github header to learn what to do.

Github and Wiki

In your project, you have to use Git and Github as a version control system. You will create a repository that can be public or private. If you choose to use a private repository, you will need to grant your TA access to the repository. We will track your commit histories and messages. You should come up with meaningful and explanatory commit messages. Messages such as “Fixxx”, “yeaaah, it is working” or “I am stuck” will not be helpful. For instance, it will make it difficult for you to browse your history and find the time you first introduced a feature. Good commit messages can save you tons of time while finding a particular commit.

Show us you can use Github almost proficiently. Create branches/forks, pull requests, and issues. Reference your issues from your commits, pull requests, and vice versa. If you become good at them, you will be ready to contribute to any Open Source repository you like, outside the course. It could be your favorite Game Engine, Programming Language or Operating System. Given you have enough domain and Git knowledge, nothing can stop you from contributing to an Android project, for instance. Of course, you should follow their contribution guidelines. Contributing to an Open Source project will shine in your CV and help you gain visibility.

We also expect you to use Github Wiki pages so that you can manage your project and keep track of progress. Please read this first, then head on over to the wiki and update your group's "First Page" (adding the title, group name, names of groups members, etc). Remember to keep logs of your Group meetings there too, and, most importantly, create and maintain your Personal Log page. Don't forget, keeping a record of what you personally do each week is a minimum course requirement!