Software Engineering Final Project

1. Restructuring or Sprint Planning (10 pts, due 08/12/2023)

- **Restructuring:** If you experienced an extreme situation in the previous project planning and your group project fell apart, consult your instructor, and read the following restructuring stage:
  - Important Note: If you do not make the corrections in this step, you may get a very low score as it will not make sense in the next steps.
  - Create a 1-page project poster which outlines your project. Use clear and understandable schematics, screenshots, and figures to explain how to use your program.
  - **Project Info**
    - Intended use of the system and its overall functionality. What is the main purpose and highlight of your project? Please write these sections as items.
    - Main components of the system. Describe each component of the system. And define how to connect each component to get the main system.
  - **System requirements - Product backlog**
    - User story and sub-stories (ex: user-login-story, user-ticket-purchase-story, admin-login-story, admin-view-monthly-report-story).
    - Convert epic stories to short stories.
    - Provide priority of these stories as low, mid, high.
    - Provide a project timeline according to sprint and presentation dates.
  - Go to sprint planning stage.
- **Sprint Planning:** If you can continue your project without any problems, you have moved on to the sprint planning stage:
  - Conduct a meeting attended by all members of the group and plan for scrum purposes.
  - Select roles: scrum master and development team.
  - Create sprint plan: sprint backlogs and task table.
  - Create a burn-down chart for delivery time estimation according to sprint plan.

2. Weekly Sprint Reports (3x10 pts=30pts)

- Sprint Report I deadline: 15/12/2023
- Sprint Report II deadline: 22/12/2023
- Sprint Report III deadline: 29/12/2023
- At the end of each sprint, submit a sprint report (PDF, no page limit) containing the following:
  - What functionality does the system have at the end of this sprint?
  - List the user stories that you successfully implemented during this sprint.
  - Did you end up making any changes to any of these user stories? Did you break down further any the user stories? Did you identify any new user stories during this sprint and, if so, did you add them to the product backlog or decide to implement them right away? Explain
  - What are the "lessons learned" at the end of this sprint? What would you do differently next time? Explain
  - Provide an updated numbered list of all user stories yet to be implemented; indicate pre- and post-conditions.
  - Use burn (up or down) chart to present current and predicted progress.
  - If this is not the last sprint:
    - Given the current functionality of the system and considering the pre- and post-conditions, identify a subset of user stories to be implemented during the next sprint.
    - Be sure that the cumulative size of the selected user stories is about 1/3 of the size of the full backlog.
    - Describe the functionality that your (partially implemented) system will have at the end of this sprint.
  - If this is the last sprint:
    - Are there any user stories left unimplemented in the backlog?
    - Are there any new user stories that you would consider adding to the backlog?
    - List these user stories and explain them.

3. User manual and Project demonstration (50 pts, due last lecture week)

- User manual: Submit the user manual document (PDF, 2 pages minimum, not including the screenshots) as described below:

    - Detail all necessary steps needed to deploy/install your system. Provide all necessary technical specifications
    - Explain the main features of the system to a potential user who may not be familiar with it.
    - Provide a walkthrough for the main scenario of using your system; include screenshots as necessary.
    - Provide walkthroughs for at least two additional scenarios with additional/alternative functionality; include screenshots as necessary.
    - Use clear and understandable schematics, screenshots and figures to explain how to use your program.
- Project Demonstration
    - All software developed within this project must be successfully demonstrated in class. Each demonstration must be accompanied by a brief presentation explaining the nature/specifics of the project.
    - Grading and teamwork survey
    - Present GitHub contribution report.
    - All deliverables will be graded because of the work of the entire team. However, individual students may receive different grades based on the degree and quality of their involvement in the project. To facilitate the objectivity in grading, each student will be required to complete one or more confidential survey about the involvement of other members of his or her team in the project. These surveys will be strictly confidential. Students who fail to complete this survey will receive a grade of 0 for the entire course project.

4. Project Test Report (10 pts, due one day before final exam or it will be decided later)

Prepare your test report for the projects (at least 2) assigned to you. (max 5 pages per test)

- Testing will be done in terms of GUI, usability, and functionality of the software.
- Write project information.
- Write your test plan and strategy.
- Present test results and errors
- Write review comments.
- Score GUI out of 10
- Score usability out of 10
- Score functionality out of 10
- Calculate overall score using.
    - $\text{Score}_{\text{general}} = \text{Score}_{\text{GUI}} \times 0.2 + \text{Score}_{\text{usability}} \times 0.2 + \text{Score}_{\text{GUI}} \times 0.6$