

# Software Engineering Course

## Project specification

### Introduction

- In this project you will collect the requirements, design and implement a non-trivial software system. You will practice the concepts you learned during the course.
- This document states the different project phases and their details.
- The project's 3 phases are: Requirements, Design, and implementation and testing.
- Your project customer (whom you can check requirements with) and coach is your TA.

### Project Logistics

- 1- Students will form groups of 2~3 members.
- 2- Your team will register their names with the TA and **you CANNOT change teams** after registration.
- 3- Students will pick 1 from 3 available projects or propose their own with professor's approval.
- 4- Every topic will be assigned to three teams at max.
- 5- Academic honesty is assumed. All work submitted must be original and written by your team (Not copied from students, the net, outside sources). Professional conduct and practice is essential in your career. **Plagiarism will be severely penalized.**

### Project Phases:

Phase	Deliverables	Deadline	Mark
Phase 1-a	Initial SRS Document (Introduction and Requirements sections of CIE203-2016-Phase1-SRS-Template-v1.0)	8 March	1
Phase 1-b	Final SRS Document	15 March	3
Phase 2-a	Initial SDD Document	22 March	1
Phase 2-b	Final SDD Document	29 March	3
Phase 3-a	Initial Partial Implementation	5 April	1
Phase 3-b	Final Partial Implementation	12 April	3
Phase 3-c	Project Presentation	19 April	3
Total Project Grade			15

### Phase 1: SRS Document

- Project description is included in a separate document.
  - TA will act as your product owner (reference for any questions).

- Your role is to understand and specify the key features / requirements of the product.
- Think about the missing details and discuss them with TA.
- Ensure that you fully understand what the product owner needs.
- Do not add any extra major features on your own. **It is beyond the scope!**

### Details

- Each team will read and understand the given project system description.
- List **all the functionality** in the project with a number from 1-5 that express the complexity of the functionality. 1 is easy and 5 is complex.
- Make sure to think in any **missing details**, further related sub-features and discuss with the TA if needed.
- For each functionality, a Use Case Table will be provided that describes the functionality in details. Document should contain all the functionality covering the requirements.
- You should determine the related non functional requirements and explain them. We expect at least **2 x team size** non functional requirements.
- Use case diagram and any further points in the template should also be filled.
- You will **submit** work on **2 stages**.
  - In the first stage, you will submit document with **ONLY introduction and requirements section of the SRS template CIE203-2016-Phase1-SRS-Template-v1.0**. Your TA will give you quick feedback on your work.
  - Then, in stage 2, you will submit the updated document considering all the mistakes. Don't hesitate to ask TA / Visit him in office hours.
- Check the sample SRS document provided. (More helpful examples are also included).

### Phase 2: SDD Document

- In this phase we will work on the Software Design Specification Document.
- Details will be published as needed.

### Phase 3: Implementation & Quality Assurance

- The target of this phase is converting the design to implementation while asserting on quality perspective.
- Details will be published as needed.

### Presentation

- Each team will present their work in 5 to 6 minutes, covering the following topics:
- Details will be published as needed.

# Grading

## Phase 1: SRS Document

### Draft SRS Submission [1 marks]

- Serious work will be given **full mark**, regardless of some mistakes.
- Shallow or incomplete work will get **half mark**
- Work that doesn't make sense or over trivial will get **zero**.

### Final SRS Submission [3 marks]

- **0.25** mark - report format, structure, language and grammar
- **0.25** mark - Software Purpose, Scope, and Definitions.
- **1** marks - functional
  - 1 grade for correctly converting problem statement requirements. Students shouldn't miss any required operation
  - 0.5 grade for not elaborating on the requirements, listing the **missing details** for them.
  - -0.5 for going out of scope and introducing un-releated major features.
- **0.25** mark - non-functional requirements
  - **At least 2 x team size** non functional requirements
- **0.25** mark - use case diagram (model)
  - -0.3 for incorrect include / extend relationships
- **1** marks - use case tables
  - **Should cover ALL use cases of the system**
  - Tables should be very clear. -0.5 for bad flows.