EECS4312 Software Engineering Requirements
Course Project (100 points), Version 1

Instructors: Song Wang
Release Date: Nov 3, 2020

The group software project has been designed to offer the opportunity to apply the techniques of Requirements Engineering taught in class. Each software project is developed through the interaction of a PM (product manager) team and a Dev (developer) team. The PM team can apply techniques of requirements elicitation, negotiation, modeling, and formal specification (if needed) to generate the SRS. The ultimate outcome is a software requirement specification (SRS), developed by the PM team and a software prototype, developed by the Dev team. The Dev team implements the software prototype based on the SRS provided by a PM team. Consequently, each student is involved in two projects, as a PM for one and as a Dev for the second project.

For each project, the PM team will hold meetings to develop SRS. Similarly, the Dev team will hold meetings with PM team to understand SRS, design the software and finish the prototype implementation. Additionally, the PM and Dev will meet several times during sessions of requirements elicitation, design negotiation and prototype demonstration.

Policies

- You are required to work in a group. Team members in a team will have the same marks regarding the quality of your submission (90% of this project). The rest 10% will be distributed based on the contribution of each member in a group.
- You are required to develop and maintain your project (i.e., both documents and source code) in GitHub.
- When you submit your work, you claim that it is solely your work. Therefore, it is considered as an violation of academic integrity if you copy or share any parts of your code during any stages of your development.
- Your (submitted or un-submitted) solution to this assignment (which is not revealed to the public) remains the property of the EECS department. Do not distribute or share your code in any public media (e.g., a non-private Github repository) in any way, shape, or form. The department reserves the right to take necessary actions upon found violations of this policy.
- Emailing your solutions to the instructor or TAs will not be acceptable.

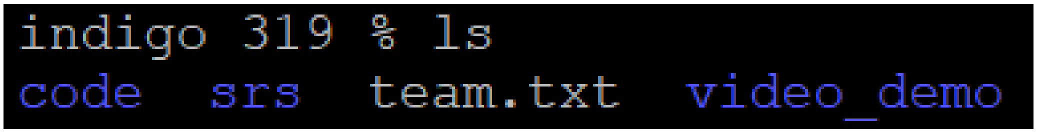
Project Phases & Deadlines

The tasks in this project can be divided into the four phases and take 90% overall marks, which are shown in Table 1. Detailed tasks in each phase are as follows:

- **Team Init**: Each team should create a repository in GitHub with the initial structure showed in Figure 1. The repository will be shared with all team members and the instructor (https://github.com/waooog). Each member should make one commit to append his name into 'team.txt';

Table 1: Project Phases

Tasks	**Marks (%)**	**Deadline**
Team Init	1	11:59 PM, Nov. 5, 2020
Project Selection	5	11:59 PM, Nov. 5, 2020
SRS Doc	51	11:59 PM, Nov. 14, 2020
Implementation	43	11:59 PM, Dec. 5, 2020



```
indigo 319 % ls
code  srs  team.txt  video_demo
```

Figure 1: Structure of your repo (**'code'** will be the folder of your source code, and **'srs'** will be the folder of your software requirement specification, **'video_demo'** will be the folder for the demo of your project prototype).

- **Project Selection**: We provide four projects for all teams to choose from which is in Section 0.1. To make sure every project has at least two teams, please select at least 3 projects you want work on as PM team and 3 projects that you want to work on as Dev team. Please send your selections to your instructor before the deadline via email with title **"EECS4312 Final Project Selection"**. Please also indicate who are your team members in this email.
- **SRS Doc**: You are asked to write a SRS doc (template is in Lab0) and Use Case diagrams. To elicit requirements, feel free to book meetings with your client, i.e., your instructor.
- **Implementation**: You are asked to implement the software with the 'SRS Doc' provided by a PM team, feel free to schedule meetings with the PM team to discuss vague requirements during the development process.

Submitting Your Work and Marking

Instructor will git clone your repo (will not modify your project) and collect specified deliverables for marking.

Your project deliverables will be marked by both instructor (weight: 30%) and other teams (weight: 70%). Marking scheme for assessing the quality of deliverables will be posted on the course webpage.

The deliverables will take 90% marks of this project and the rest 10% marks will be distributed based on the contribution of each member in a team. The distribution will be made by your team.

Your Role as a PM (Deliverables – PM)

You will be assigned to a PM team and work on a project agreed by the course instructor. Your team will develop a software requirements specification (SRS) for the project you selected. You will describe the problem for which a software system is to be designed, and comment on the prototypes done by the Dev team. The following are the deliverables for your PM team:

- **Create Software Requirement Specification (SRS)**: This should outline the project requirements. This document should state the scope of the project, functions that users may expect the system to have. Make sure the project is suitable for completion in 3-4 weeks. Remember that it is not a competition; do not aim to destroy the Dev but give an honest set of requirements. (Marks will be deducted for a project considered too easy or too difficult.) The marking scheme will be posted on the course webpage.
- **Create Use Cases**: Identify the main actors in the system. For each of the actors, please identify its use cases under different scenarios and draw an use case diagram for each use case. Please use https://app.di-

agrams.net to draw the diagram. Put all your diagrams in one doc under the folder of ‘srs’ in your repo.

- **Grade SRS developed by other teams:** A project will have multiple teams working on it, you will have the opportunity to comment and grade the SRS (including both SRS doc and use cases) created by other teams that worked on the same project with you. Marking scheme for assessing the quality of the SRS version 1.0 document will also be posted on the course webpage.
- **Grade Project Prototype:** The final implementation and a demo of the preliminary project (5 minutes video) will be provided by the Dev team. Marking scheme for assessing the quality of the project prototype will be posted on the course webpage.

Your Role as a Dev (Deliverables – Dev)

You will be assigned to a Dev team and work on a project (different from the one you work as a PM) assigned by the instructor. Your team will respond to a SRS prepared by the PM team. You will then collaborate with the PM team to clarify vague requirements and implement the project prototype. You can analyze and specify the requirements for the system desired by the PM team, through an iterative process of prototype design and evaluation in collaboration with the PM team. The implementation and project demo will be evaluated and assessed by the PM and the instructor. The following are your deliverables for your Dev team:

- **Project Prototype:** In response to the PM team’s ‘SRS Doc’, your team will implement the project and prepare a preliminary project demo (5 mins) for evaluation by the PM team and the instructor.

0.1 Alternative Projects

P1: Express Parking

The Toronto Parking Authority operates over 17,500 parking spots across Toronto. Approximately, 10,000 of these spaces are by 2,000 ticket meters and the additional 7,500 by single space meters. Currently, a customer either purchases a ticket from the machine and places it on the car’s dashboard or inserts coins into a single spot meter. Employees have to walk weekly to collect money from these 9,500 locations. Additionally, the Toronto Police Service offers parking enforcement via dispatched personnel who randomly select areas to monitor. These personnel need to check tickets or parking meters manually and issue tickets as needed. In order to minimize the number of locations to collect money, the Toronto Parking Authority wants to develop an online platform to process payment for select spaces where single parking meters are currently used. The spaces will be numbered and the registered customer can select the space number from their device online to initiate payment. Additionally, parking enforcement will have a separate login into the application which will allow them to see whether an occupied space has been paid for or manage the parking locations.

P2: School Attendance System

We are a flourishing school board consisting of 150 high schools and approximately 154,000 students across all of our schools. We have observed an annual enrollment growth of about 1000 students per year. Our staff secretaries are overloaded with attendance work resulting from the excessive paperwork required by the current system. In the carrying out of such a repetitive task, human error is inevitable. Our school board, like most others in our area, is funded based on attendance rates so this human error has resulted in significant losses in funding. Funding lost due to false absences is about 50K annually. This is a significant dollar amount, which we wish to retain by implementing an improved online attendance system.

The proposed solution is to establish a single website such that all users are able to interact with it in the manner that they are intended to. The website will be directly related to a database server which will contain all attendance records. The website will be able to login administrative staff according to the school which they are related to, it will also take in the consideration of teachers which may be teaching at multiple schools. Supply teachers can also be given permission on a daily basis from the administration task with a temporary unique ID and password which they can use to access the system for all classes that they are teaching that day.

The website structure takes into account the situation where parents have multiple children in different schools within the same school district. It can provide a means of usability and easiness to the parent such that they are able to log in from one main website and access attendance records for all children attached to that parent. In general, the concept of the website is to keep it at a bare minimum of complication in order to reduce the level of complexity and time it would take to train the administrative staff to use the system while being able to maximize its usage.

Our current method is as follows:

- At the beginning of each school day, after the announcements, the teacher calls out the students' names and records whether they are present, absent, or late.
- After an amount of time determined by the teacher, the attendance sheet is sent to the office via a student.
- At the office, the attendance is collected by secretaries.
- The secretaries look at each attendance sheet and proceed to make phone calls and send emails to the parents of the students who were missing. This often leads to staff missing some students by mistake due to the volume of paperwork.
- If the absence was recorded by parents prior to the deadline, the secretary will take no action due to the absence or lateness.

P3: VideoCo Management System

VideoCo is a small video store with 2 stores in Toronto. VideoCo wants to expand its business and start renting videos worldwide through the Internet. They plan to establish a few warehouses scattered around the globe and a new computer system. Your help is needed in specifying this complex project.

Registered customers should be able to rent videos and have videos shipped to them. Customers should be able to return videos. Customers in Toronto area can rent and return their videos in person. Other customers will need to have videos shipped to them. Customers will be charged a late fee based on their location. To encourage repeat customers, VideoCo would like to have a loyalty program in their new system. Customers that are not internet savvy can dial an operator to pick the movies for them. Customers can pay through credit cards or loyalty points.

On an hourly basis their warehouse team would like to create a list of to-be-shipped movies. Gathered movies are delivered to the shipping team who in turn prepares the paper work to ship the movies worldwide.

P4: SmartShoppers System

ShoppersLand Inc. is the leading retail chain in Canada. We offer a large range of products including food, clothing, electronics, and pharmaceutical drugs. As a company we strive to provide the best shopping experience for our customers. We are continually trying to find innovative ways to help shoppers have a better experience in our stores.

We would like to develop an online system, i.e., SmartShoppers, to allow customers to find products in our stores with greater precision than the current system. When open the system, the user will need to specify their postal code and/or their city and province. They will then be presented with a list of stores within their area to choose from as the location they will be shopping at. There should also be a feature which lets the user save this store as their location for future use. The user should also be able to change the store location any time they wish. The user should then be able to create a shopping list by searching for products and by viewing a list of sale items at the store they specified. An algorithm should then provide the user with the list in the best order in which to find the items, starting at the front of the store. Users should also be able to click on items to see a product description, price and where in the store it can be found. The user should also be able to view a list of suggested items. This list would consist of products that others have searched for when they have also searched for items on the users shopping list.