[CS350] 2018-Fall, Team Project

Project #4. Sprint 2

Open: 2018-11-09 (Nov. 9th)
Due: 2018-11-26 (Nov. 26th)

In *Idea Proposal* (*Project #1)*, you have proposed an idea for your project and you also have devised functional and non-functional requirements to develop an application. In *SRS (Project #2)*, you have made the analysis on use cases, domain, and acceptance criteria, including non-functional requirements and preliminary user manual. After the analysis, you were getting started to develop a system following agile process, and you have shown how the basic functionalities and UI of your system work by prototyping it in Sprint #1 (*Project #3*).

Topic Selection. In Project #4, you are requested to extend (and improve) your system further. Based on your team's plan, you have to design some parts to be improved and you have to implement them. Before getting started, you will select topics for Sprint #2 as follows:

- You have three categories to choose from: (a) adding new functionalities, (b) supporting non-functional requirements, or (c) improving basic functionalities (Finalizing requirements written in SRS). The following table shows the example of those three categories on the example of "Hotel Reservation System". You should choose more than one topic among those three categories and include the content of topic selection on the Sprint2 document (e.g., 2-A. Goal)
- You can use external libraries or services for implementing new functionalities that you didn't elicit on the SRS document. Please note that on Sprint 2 document (e.g., 2-D. Tasks for Sprint2 – Description), if you use another or different public data APIs.

Adding new functionalities	
Data gathering/learning (with *ML*)	Design & gather various data for later analysis or new business opportunities E.g., Filter & analyze various data for marketing promotion
Chatbot (with *NLP)*	Perform tasks using a *Natural Language Processing* technique E.g., Reserve a room via. chatting interface
Other free topics	Select & propose a new functionality to support/improve the current version of HRS. It should include a technical idea to support your basic HRS functionalities, such as effective/quick reservation, membership management, policy management.
Supporting non-functional requirements *to improve your system quality*	
Fault-tolerance	Provide the system with failure recovery/error-handling
System modification	Make the system general and be portable, scalable
Usability	Upgrade the UI to be more user-friendly and fancy (E.g., Look & Feel, Accessibility)
Security	Provide the system with authentication, privacy protection
Other NF features	Select & propose a new non-functional feature freely.

Full implementation *to support making/modifying/canceling a reservation*	
Data management	Design the data schema, and connect your system to a DBMS
Network	Implement back-end, and make it work via. the Internet
Mobile application support	Provide mobile services with easier reservation

Report Document: You need to refine the analysis and system design to extend your system. The implemented system must be validated by reviewing the requirements, which were defined at the beginning step of Sprint #2. You need to include the following sections in your report: (no more than **20 pages,** use appendix if needed)

1. Title Page

Include the name of the document, team name, team members, date. You also prepare *Table of Contents* (including page numbers and so on) of your document.

2. Sprint Backlog

Write a quick description of the sprint goal. Then identify one or more user stories and requirements that support this goal and will be part of the sprint. Specify the tasks to accomplish the requirements, and map tasks to requirements that are covered by each task.

A. Goal

The goal of this sprint is a vision statement for this sprint, to communicate which parts of your 'product' you are going to develop.

B. User Stories for describing your topic

Elaborate one or more user stories (i.e., happy case scenario) that cover the goal of this sprint. They will show how users are going to interact with your product. Requirements can be found in some user stories, specifying the needs of users.

C. Requirements

Sort out the requirements to achieve the goal of this sprint (to improve your system). You need to classify and number the requirements with tables like the following.

Functional requirements	
R. ID	**Description**
R.F.1	The customer can make a reservation
Non-functional requirements	
R. ID	**Description**
R.N.1	The response should be delivered within 1 second.

*The IDs of requirements can be freely specified, and you can change the format of the requirements table if needed.

D. Tasks for Sprint 2

List the tasks which should be done in this sprint to develop an initial version of your system. Make a task model in the hierarchical form. Identify which requirements are achieved by which task, and map all the requirements to the task model. The following example could be your task model.

Task Model		
Task ID	**Description**	**Related Req.**
T.1	Build the development environment	-
T.2	Implement 'make a reservation' functionality	R.F.1
T.2.1	Imple. 'Retrieve the available room list'	R.N.1

*A task can be related to more than one requirements.

3. System Design

For all the updated parts, please mark the changed (added/modified) ones in red.

A. (Refined) System Architecture

Provide a blueprint of your system with the structure and the relationships between system components. Any architectural format can be used, but your architecture should include major building blocks, and interfaces between the building blocks.

B. (Refined) Overall Use Case Diagram

In order to achieve a goal(s) of this sprint, your team will have to define some additional use cases or to modify some use cases. Please draw an up-to-date overall use case diagram for this sprint. You should mark modified or added parts in red (red circles for use cases, red lines for communication lines & relationships, red texts for names & labels).

*Do not exceed 1 page for the overall use case diagram. Please include specific modules and their use cases in Appendix, if needed.

C. (Refined) Class Diagram

For this sprint, your team will also have to add or modify several classes. Identify those components and refine the class diagram that you designed in Sprint 1 to develop an improved system in Sprint 2.

4. Sprint Review *(same as Sprint 1)*

A. Achievements

Make a self-check of your achievement at the end of the sprint. You need to include (a) *task achievement* in the form of a checklist, to show how much your team achieved the original plan you made at the beginning of the sprint. You also need to include (b) *requirement validation* (testing) for each requirement listed in ***2-C*** based on the acceptance criteria of *SRS*.

Requirement validation must include 1) test input, expected output, actual output and test result for functional requirements, and 2) goal, actual output, satisfaction result, and the reason why it is satisfied or not, for non-functional requirements.

B. Demonstration

Show your implementation results with screenshots. Put remarks of requirement IDs on related screenshots. Please follow the order of your team's user story (***2-B***).

C. Development Environment

Describe the environment for development and testing. It may include programming language, used platform/framework (and its version information), desktop & server information, and so on.

5. Roles & Responsibilities

State roles and responsibilities for each member of your team in a table.

A team manager, please explain how you managed sprint 2 of your project. Explain briefly any tool, planning, strategy, schedules, and motivating method you used to manage, promote and accelerate your progress. Write your name also.

6. Acknowledgment

Include authors of this document by section.

[Scoring criteria]

- **Agile plan design:** sprint backlog
- **Completeness**: requirements, use-case diagram, user stories, sprint review
- **Consistency**: requirements, use-case diagram, sequence diagram, class diagram
- **Design skills**: system architecture, class diagram, sequence diagram
- **Implementation results**: runnable system

[Project schedule & due]: *Information of each schedule will be announced soon.*

- **Final presentation:** Nov. 29th, 7PM, N1-201 (with professor Bae) – 15 min + 5 min for Q&A
- **Final report:** Nov. 30th

[Submission files]

Please submit below files to Sprint 2 submission tab, KLMS by team.

1) **Source code:** Please submit ".txt" file that has your Github repository link or ".zip" file of your project. You may include an executable program or URL link of executable website (in case that your system is a web-based app).
2) **Sprint 2 document**
3) **A short video:** Please shoot a video of your system running to demonstrate your system.