# Project: Software Engineering Analysis – Home Delivery Platform

## Introduction

You and your fellow students want to create a new home delivery services start-up company. During the pandemic crisis you saw that a lot of shops, pharmacies, and restaurants needed to newly introduce home delivery services very quickly. At the same time, unfortunately many students have problems in finding student jobs in bars, shops, etc. Your idea is to bring these two groups of people together by offering a home delivery platform. Everybody who needs a delivery service can register and ask for a delivery. At the same time, contractors can also register and pick-up delivery orders from the system.

It should be easy and straightforward for your customers and contractors to use your services from any web-browser or by installing apps on their mobile phones. In addition, you and your colleagues should also be able to control all functions of the system through web and mobile clients.

You did a little bit of brainstorming and came to a very preliminary list of requirements for your system:

1. Customer and contractor registration and management
2. Payment through an external service provider
3. Placement of delivery orders
4. Order pick up by contractors
5. Tracking of the delivery process and notification of all people about the progress
6. Potentially pre-ordering of delivery services some time in advance
7. Contractor rating system
8. Check revenue of your company, maintenance cost and status of your contractor network
9. Check growth of customer base and revenue per customer
10. Analyze customer satisfaction and collect suggestions for improvements

You certainly need to continue requirement collection at a later stage. You decided to use an Agile software development process and selected a Scrum framework (which will be discussed in the lecture in January).

As there are a lot of companies trying to get into this market, you need to act quickly now. You have a total of nine weeks to work on this project. You understand that you need to find investors who are funding the actual development of your software solution. **Therefore, you will need to present your project results at the end of the semester term during your exercise on 05.02.2021 and 08.02.2021, respectively. In addition, you need to submit a written report afterward until 19.02.2021. Please note that the submission is final once you submitted your project report and no more changes can be made afterward.**

The objective for this project is to collect a clear understanding of the requirements and develop use case scenarios and UML diagrams to describe your planned software product. In addition, you should build user interface prototypes for relevant parts of the functionality.

**Please note that the project scope and expected deliverables might be updated during the course of the project (as usual).**

## First 3 Weeks of the Project (including year-end break)

The project will start officially in your exercise on 18.12.2020 or 21.12.2020, respectively.

In the first three weeks until 08.01.2021 / 11.01.2021, you need to collect additional requirements (e.g., as Snowcards) and refine the requirements to build a list at a reasonable level of granularity. Assign priorities to your requirements (importance = high / medium / low).

Each team member should then select at least one requirement and perform an object-oriented analysis as practiced in previous exercises. The outcome of this analysis should be the following sets of diagrams:

- ☐ Use case diagram
- ☐ Activity diagram
- ☐ Class diagram
- ☐ Sequence diagram

Please create a table clearly indicating who worked on which parts of the results and when the team meetings took place (including a list of attendees for each meeting).

In the exercise on January 8th / January 11th, you can present your diagrams to your tutor in order to double-check the quality and correctness of your results.

## Following Weeks of the Project

In January, the agile Scrum framework will be introduced in the lecture. In the following, you will adopt your way of working together in the team to Scrum.

This requires that you turn your list of requirements into backlog items. Please note that the purpose of a backlog item is to take a requirement description (in form of a use case description or Snowcard) as input and build an analysis model of your software capability as result. For complex requirements, you may consider breaking it down into smaller sub-items.

The results need to match the following definition of "Done" (checklist):

- ☐ Description of the requirement in form of a Use Case
- ☐ Categorization of the requirement (functional / non-functional)
- ☐ Business value of the corresponding functionality (low, medium, high)
- ☐ UML Diagrams
    - Use case diagram
    - Activity diagram
    - Class diagram
    - Sequence diagram

  Depending on the type of requirement, not all these diagrams may be applicable. If a diagram is omitted, describe why.
- ☐ For UI related functions: UI prototype
- ☐ Detailed documentation (e.g., table) about who worked on the item and what has been done during the sprint by whom (provide percentage break-down of workload to team members).
- ☐ Overall quality of the documentation meets usual standards.
- ☐ The results have been reviewed and accepted by another member of the team (tester). It needs to be documented who has performed the review.

In the following weeks, you should conduct sprint review and sprint planning meetings as part of the exercises. You start with a sprint review meeting where each team member presents the results of the previous sprints. The product owner then decides whether the backlog item is Done (according to the definition above) or whether it needs to be moved back to the product backlog.

Afterward you conduct a retrospective meeting where you investigate the quality of your process and how it can be improved. You also judge about whether previous improvement ideas have been successfully implemented. You then need to plan definite improvements for your process. Please document your findings in the project report to convince your potential investors about the quality of your software process. Ensure that your Scrum process is improving and that the sprints are carried out successfully.

Finally, you perform a sprint planning meeting where you select items for the next sprint backlog. Break down these backlog items into tasks and assign these tasks to individual team members. Ensure that you have a clear understanding of the expected deliverables for each of these tasks. The team then estimates the time it will take to complete each of these tasks. Adjust the plan and iterate as needed.

During the sprints, the product owner collects additional requirements and adjusts priorities if necessary. The other team members work on completing their tasks and produce the required artifacts. The team should conduct frequent "daily" Scrum meetings.

The schedule will then be as follows:

| **Meeting** | **Tasks** |
|---|---|
| 1 | Engineer requirements, develop product backlog and estimate backlog items |
| 2 | Sprint planning #1 |
| 3 | Sprint review #1, retrospective, sprint planning #2 |
| 4 | Sprint review #2, retrospective, sprint planning #3 |
| 5 | Sprint review #3, retrospective, sprint planning #4 |
| 6 | Sprint review #4, retrospective (presentation and report review) |
| **5.2. / 8.2.2021** | **Project presentation** |
| **19.2.2021** | **Submission of the written project report** |

## Evaluation

The evaluation of the project regarding the pass / fail rating will be based on the following three aspects:

1) Documentation of the project: Write protocols of your meetings (e.g., sprint planning, retrospective, and sprint review) and the sprint progress / success. Which backlog items have been selected for the sprint? Who has worked on which backlog items and has produced which results and when? Who attended which meeting? These individual reports need to be combined into a single project report which must be submitted at the end of the project. Ensure that the report follows usual standards for such documents (e.g., quality, length, structure, formatting). **The final project report needs to be submitted as PDF document until 05.03.2021** (including a two-week extension due to the Corona situation).
2) Content of the project and quality of your software process: Ensure to work on enough relevant capabilities (requirements, backlog items, etc.) of your application to impress your investors. **It is expected that the final DIN-A4 project report contains 80-120 pages.**
3) Presentation: In the exercises on **05.02.2021 / 08.02.2021**, the results of the project will be presented in a short presentation given by all team members.

## Contents and Style of Project Report

The project report should contain the following sections:

1) Title page (including project title, names, and matriculation numbers of the team members)
2) Declaration of authorship (see below)
3) Table of contents
4) Introduction (optional)
5) Tabular list of all backlog items / requirements (see below)
6) Main part consisting of a collection of all backlog items according to the definition of "Done". This should include Use case diagrams, activity diagrams, class diagrams and sequence diagrams, and UI prototypes. Not all of these parts may be relevant for each backlog items. Write a brief explanation in case that you decide to skip an item.
7) Summary: Brief summary explaining your feedback regarding the project
8) Appendix: Protocols of your meetings

Regarding the **style and formatting** of the report, you can optionally use Magic Draw's report wizard to create an initial template containing all your diagrams. Otherwise, you should follow the style of a typical bachelor / master thesis report. A style template for LaTeX can be found at:

https://www.latextemplates.com/template/masters-doctoral-thesis.

Alternatively, you can also use other word processing software, e.g., Microsoft Word. If you do so, please use/create a formatting style similar to the suggested template.

Regarding the **declaration of authorship**, you can use a short text similar to the following:

*Declaration of Authorship*

*We hereby certify that the project report we are submitting is entirely our own original work except where otherwise indicated. We are aware of the University's regulations concerning plagiarism, including those regulations concerning disciplinary actions that may result from plagiarism. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.*

The text needs to be followed by your team member's names, matriculation numbers, your signatures, and the signature date.

The **tabular list of all backlog items** should contain the following columns:

1. Main actor for the use case (customer, contractor)
2. Title of the backlog item
3. Categorization (functional, non-functional)
4. Business value (low, medium, high)
5. Sprint or week when this backlog item was worked on
6. List of team members who worked on which part of this item and to which percentage (in brackets)

You should sort the table by the main actor for the use case. You may also split the table in order to improve the page fit.