Cairo University
Faculty of Computers and Artificial Intelligence

Software Design and Architecture

Project Description

2020/2021

Version 1.0

Project Team

Staff: Dr SohaMakady **s.makady@fci-cu.edu.eg**

TAs: Eng Hassan Mourad **h.morad@fci-cu.edu.eg**
Eng Mohamed Samir **m.samir@fci-cu.edu.eg**

Project Description – 2021/2022

Cairo University, Faculty of Computers and Information

Introduction

- In this project you will design and implement a non-trivial software system. You will practice the concepts you learned during the course.
- Project will be based on agile practices with at least 3 sprints
- In each sprint we will focus on some requirements from the requirements backlog. We will design and implement these requirements
- In each sprint each team is required to deliver the following
    - Software design specification document contains the following
        - Proposed architecture: Should include a subsystem decomposition, and a component diagram showing the interfaces of the different components.
        - Proposed class diagram
        - sequence diagrams for the **most complex** scenarios. The submitted sequence diagrams should be **2 x the size of the team**, where each team member would be responsible **for submitting two sequence diagrams**.
        - Check the SDS document with the project description
    - Sprint document
        - Meeting minutes for the sprint starting meeting.
        - Meeting minutes for the sprint standup meetings.
        - Meeting minutes for the spring retrospective meeting.
        - Trello board screenshot
        - Git repository for the developed source code.
        - Check the Sprint document with the project description
- Project consists of 3 phases, each phase represents a sprint/iteration.
- Your project customer (whom you can check requirements with) and coach is your TA.
- For more information about the different sprint terms mentioned above, and the overall agile software development process, please refer to this link https://www.mountaingoatsoftware.com/agile/scrum/resources/overview

Project Logistics

1. Students from the same lab/TA will be divided into groups; each group consists of 3 members.
2. Your team will register their names with the TA and **you CANNOT change teams** after registration.
3. Academic honesty is assumed. All work submitted must be original and written by your team (Not copied from students, the net, outside sources). Plagiarism will be penalized.
    - Soon, you will be our colleague and we will be proud of you.

- Professional conduct and practice is essential in your career.

Project Phases:

Phase	Deliverables	Deadline	Mark
Sprint 1 progress update during the labs	**Show the TA DRAFT class diagram, and one DRAFT sequence diagram from each team member to get feedback from the TA on your progress, and help you with any difficulty.** • **This phase is MANDATORY. ALL team members should come to the labs with their class diagrams/sequence diagrams.** • **Not showing up to the "Sprint update" would deduct 20% from the sprint grade.** • **Showing up to the "Sprint update" would not result in losing grades for the submitted DRAFT diagrams. You can still enhance them before the final deadline based on the feedback session with your TA.**	During each team's course lab that would be held from Nov. 13, 2021 to Nov. 16th, 2021 Late submission is not allowd	
Sprint 1 submission on blackboard	**Design and implement Sprint 1 user stories (mentioned below)** Submit all the required deliverables (mentioned above)	Nov. 23, 2021 Late submission is not allowd	

Project overview

In the current days the transportation technologies are growing rapidly. Therefore, in this project we are going to develop an application that helps users to communicate with car drivers to transport users to any area.

In this project you are required to develop the requested functions w.r.t the SOLID principles. You should consider using design patterns to adhere to the SOLID principles.

Your design also should adhere the OOP concepts. So the basic unit in your class should be the "class".

You should think about an efficient design that will be suitable if the requirements are extended. Also you should think about a portable design to be used in any other user cases.

You are free to choose any programming language that you want. However, The design concepts in the labs will be explained with Java.

The current functions will be exposed as a normal member functions part of a class. In the next sprint we are going to expose these functions as web services.

Requirements backlog

1 The admin user should be able to verify driver registration. So the admin should be able to list all pending driver registrations and verify any pending driver registration.

2 The user should be able to register to the system. The user should provide username, mobile number, email (optional), and password. If the user is going to register to the system as a driver so the driving license and national id should be provided. The user should be able to login into the system once the registration is completed. If the user registers as a driver, so the user should be able to login into the system once the admin user verify the registration

3 The user should be able to request a ride given a source and a destination. For a simplicity user can enter the source area's name and the destination area's name.

4 The driver should be able to add some areas to get notification when any ride is requested and one of these areas is added as the source area. These areas will be called as "favorite areas".

5 The driver should be notified if any new ride is requested from any area added as a "favorite area" to the driver.

6 The driver should be able to list all rides with source area within one of the driver's favorite areas. The driver should be able to suggest a price to this ride and notify the user with this price. Each price suggestions is called an "offer"

7 The user should be notified if any new price is added to the requested ride. So the user should be able to list all ride offers and check the driver details for each offer.

8 The user should be able to select a specific offer for the requested ride. By this selection the corresponding driver (who put the selected price) should be notified that the user accepts the suggested ride price.

9 The system should calculate the distance and estimate time arrival (ETA) with the help of google maps API

10 The system should track number of calls to google maps API if it's exceeded specific number of calls. The system should change the way to calculate ETA by using harvesine distance to calculate the distance between the source area and the destination area then divide this distance by the speed (Assume 60 KM/H speed).

11 The driver should be able to end the ride once the ride is finished and the ride should be added to the rides history. The ride cost should be added to the driver balance.

12 The user should be able to rate any completed ride requested before by the user.

13 The driver should be able to announce any new ride. To announce a ride the driver will provide the ride details (Title, description, ride source, destination, time to leave, and cost).

14 The user should be able to list the rides announced by the drivers and register to any ride. Once the user register to a specific ride the driver should be notified.

Sprint 1 stories

1 The user should be able to register to the system. The user should provide username, mobile number, email (optional), and password. If the user is going to register to the system as a driver so the driving license and national id should be provided. The user should be able to login into the system once the registration is completed. If the user registers as a driver, so the user should be able to login into the system once the admin user verify the registration

2 The admin user should be able to verify driver registration. So the admin should be able to list all pending driver registrations and verify any pending driver registration.

3 The user should be able to request a ride given a source and a destination. For a simplicity user can enter the source area's name and the destination area's name.

4 The driver should be able to add some areas to get notification when any ride is requested and one of these areas is added as the source area. These areas will be called as "favorite areas".

5 The driver should be notified if any new ride is requested from any area added as a "favorite area" to the driver.

6 The driver should be able to list all rides with source area within one of the driver's favorite areas. The driver should be able to suggest a price to this ride and notify the user with this price. Each price suggestions is called an "offer"

7 The user should be notified if any new price is added to the requested ride. So the user should be able to list all ride offers and check the driver details for each offer.

8 The user should be able to select a specific offer for the requested ride. By this selection the corresponding driver (who put the selected price) should be notified that the user accepts the suggested ride price.

9 The system should calculate the distance and estimate time arrival (ETA) with the help of google maps API

10 The system should track number of calls to google maps API if it's exceeded specific number of calls. The system should change the way to calculate ETA by using harvesine distance to calculate the distance between the source area and the destination area then divide this distance by the speed (Assume 60 KM/H speed).

11 The driver should be able to end the ride once the ride is finished and the ride should be added to the rides history. The ride cost should be added to the driver balance.

12 The user should be able to rate any completed ride requested before by the user.

Evaluation Criteria

1. Properly working functionality as per the sprint requirements.
2. Quality of project configuration (i.e. task management, version control, SDS documentation)
3. Consistency between the various submitted system models (i.e., the class diagrams, sequence diagrams, architecture).
4. Consistency between the submitted system models, and the working product.

Policy Regarding Plagiarism:

Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.

١ تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
٢ ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
٣ أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
٤ قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
٥ إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
٦ فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.