German International University
Dr. Amr Desouky
Eng. Ayman Iskandar

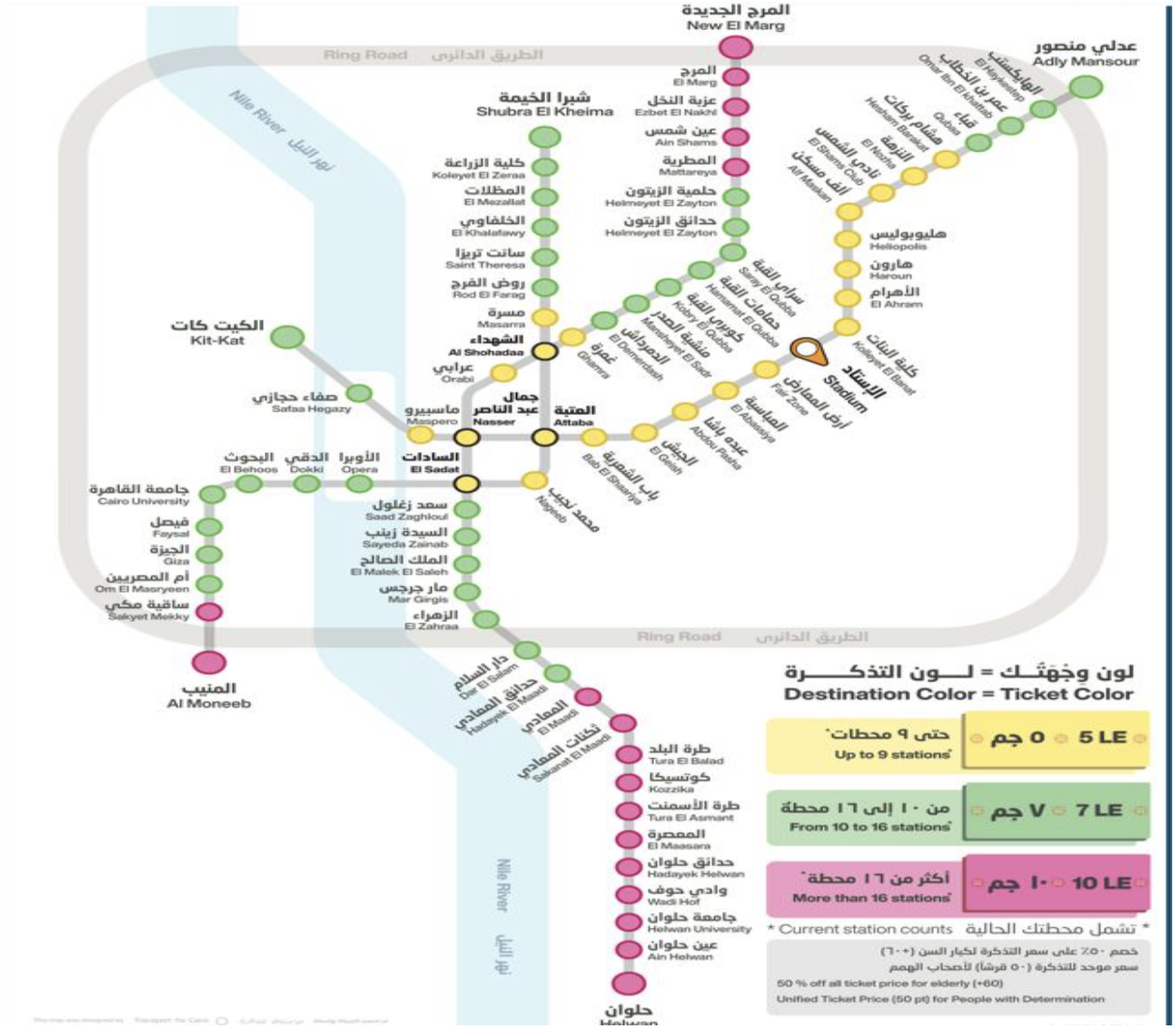
last update March 19 12:00 pm

Software Engineering
Software Project II - Cairo Metro System

Due Date: Friday March 31st 2023 11:59 PM

1. Introduction

Your team is tasked with building a ticket reservation system for the Metro system of Cairo. The system will be built across three sprints. In this sprint, your team will understand and prioritize the requirements to build the product and sprint backlogs. In addition, your team is requested to design and build your database tables.



2. Tools & Technologies

The tools & technologies required to complete this sprint are:

1) Trello (https://trello.com/)
2) Diagrams (https://app.diagrams.net/)
3) Postgresql (https://www.elephantsql.com/) - *for transactional data*
4) Mongo (https://www.mongodb.com/) - *for streaming data*
5) Swagger (https://swagger.io/)
6) Draw.io (https://draw.io/)

2. Requirements

The system shall support the following roles:

1) User
2) Senior
3) Admin

The system shall support the following features:

1) A user can register using an OTP strategy (email-sendgrid or sms-twilio)
2) A user can purchase monthly, quarterly, or annual ticket subscriptions for one-to-many zones
3) A user can view their ticket subscriptions
4) A user can view their upcoming rides
5) A user can create “virtual” rides
6) A user can view their completed rides
7) A user can request a refund for a future-dated ticket
8) A user can check the pricing of a trip by specifying origin and destination points (autocomplete)
9) When purchasing a ticket, the system should indicate full ticket price, route, and transfer stations
10) A user can upload ID image requesting “Senior” role to benefit from ticket discounts
11) An admin can create/update/delete stations
12) An admin can create/update/delete routes
13) An admin can approve or reject refund requests
14) An admin can approve or reject senior requests (manual verification or OCR automated)
15) An admin can create/update/delete/update route pricing
16) An admin can create/update/delete/update schedule times for all routes
17) A user can see an interactive map simulation of all trains waiting & moving across stations in real-time.
18) More requirements TBD

One of your tasks this sprint is to create user stories from the requirements and prioritize them in a product backlog. In addition, you are requested to create 2 sprint backlogs that show which user stories you plan on completing in a given sprint. For example, if we take the first requirement:

"A user can register using an OTP strategy (email-sendgrid or sms-twilio)"

This requirement can translate to the following user stories:
1) "Create a *users* database table to store all registered users"
2) "Create an *otp* database table to store OTP codes"
3) "Create a server-side endpoint to handle user registrations"
4) "Create a server-side endpoint to send OTP code"
5) "Allow OTP endpoint to accept debug HTTP header to return OTP in response body instead of sending an email"
6) "Create a UI screen to allow users to enter registration information"
7) "Create a UI screen to allow users to enter OTP code"
8) "Create a UI screen to allow users to request new OTP code"

A user can create "virtual" rides

This requirement can translate to the following user stories:
1) "Create a *rides* database table to capture all past, current, and upcoming rides"
2) "Create a server-side endpoint to handle user virtual ride requests"
3) "Create an interactive UI that shows a rider moving across stations until their final destination"

Each user story should be sized/estimated accordingly using the following scale:

1) high - 5 points
2) medium - 3 points
3) Low - 2 points

* A given sprint should not exceed 50 user story points.

3. Tentative Project Schedule

Sprint #	***Tasks***	***Due Date***
0	*- Formulate teams*	*March 20th 11:59 pm*
1	*- Software Architecture* *- UML diagrams* *- Product & sprint backlogs* *- Database table scripts* *- Swaggers*	*March 31st 11:59 pm*
2	*- Write backend server-side code*	*May 5th 11:59 pm*
3	*- Write frontend user interface code*	*May 26th 11:59 pm*

Project Evaluations: TBD

4. Architecture

You are all experienced seniors with sufficient experience and exposure to developing software using various architectural styles. For this reason, I will not stipulate *how* you should develop this project. Whether you choose a Monolith, Microservices, GraphQL, Event-Driven, Websockets, etc. is completely up to you. However, your design must be justified.

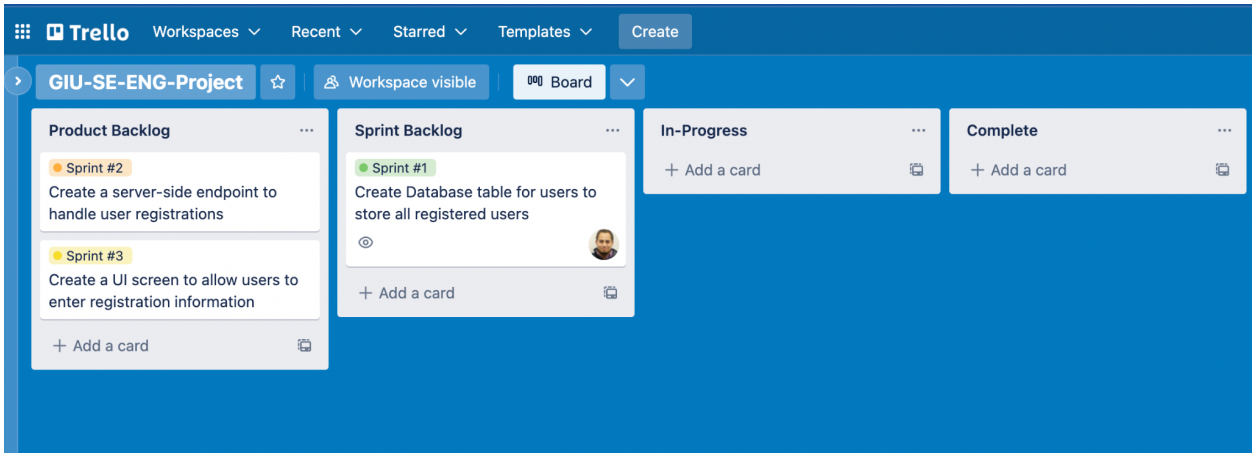
4. Services

Each team should define the services they deem necessary to fulfill the requirements. However, it's possible a given solution could include the following services:

1) Auth
2) Users
3) Scheduler
4) Routes
5) Pricing
6) Rides
7) Subscriptions
8) Security
9) Location

5. Trello

Use trello to create a single (public) board with the following swimlanes to manage your project:



Product Backlog: This swimlane is meant to capture all the user stories that makeup the system.

Sprint Backlog: This swimlane is meant to capture all the stories that will be worked in the active sprint. All stories in this swimlane should have the same sprint # label and assigned to someone from the team. If a story was not completed by the end of a given sprint, you should add a new label indicating the new sprint the story has been slated for but also keep the original one.

In-Progress: Once the sprint has started, move your assigned ticket to the "In-Progress" swimlane. If you have multiple stories assigned in a particular sprint, you should only have one in-progress. Once it's completed, move it to "Complete" and pull a new story/ticket back into "in-progress".

Complete: This swimlane indicates a story is feature complete and thus done.

6. Sprint #0 Deliverables

Each team can have up to 4 individuals; cross-tutorial teams are permitted.

Team submission link: TBA

7. Sprint #1 Deliverables

Your final submission for sprint #1 will consist of:

1. Architectural Diagram PDF
2. Link to Trello board
3. PDF with analysis (use-case/sequence diagram for each major flow)
4. SQL script to create & hydrate all tables
5. ERD Diagram (can auto-generate)

The submission link is: TBA