17-313: Foundations of Software Engineering

Fall 2019 Midterm Exam

Claire Le Goues and Michael Hilton

Name: ______________________________

Andrew ID: ______________________________

Instructions:

- Not including this cover sheet, your exam should have 7 pages in total: 7 pages of questions, and 1 page of appendix. Make sure you're not missing any pages. Write your full name and **Andrew ID** on at least this page, if not all others.
- Most questions in this exam refer to a scenario, described in the appendix. We encourage you to read it first. You may detach the appendix; the appendix will not be considered in grading, thus keep your answers on the question pages.
- Write concise, careful answers. Short and specific is much better than long, vague, and rambling, and grading will reflect this. You have enough time to write clear and readable responses. Bullet points and phrases are acceptable. Spend time to consider how best to present your answers, including citing examples to make your points more concretely.
- Clearly indicate and write your answers in the space provided below each problem. We cannot give you points for answers we cannot find or read.
- The exam has 4 multi-part questions with a maximum score of 70 points. The point value of each problem is indicated. We planned the exam to allocate approximately one minute per point.
- If you do not know the answer to a question, you may leave it blank and receive the indicated number of "No-nonsense" points (abbreviated *NN* for the rest of this exam). This may be an improvement over the zero (0) points that may be awarded an incorrect answer. The number of available *NN* points varies per question. If you start to answer a question and then change your mind, you may invoke this rule by crossing out your answer and prominently writing "LEAVING THIS BLANK." We will not read partial answers to questions on which you invoke this rule.
- You may consult one sheet of paper with notes. You may not use books, a calculator, cell phone, laptop, or any other electronic or wireless device.
- Good luck!

Question	Points	Score
Requirements	16	
Process and Metrics	14	
Architecture	22	
Machine Learning and Ethics	18	
Total:	70	

Question 1: Requirements (16 points)

(a) (8 points) *(2 NN, 1 per story)* One way to evaluate the quality of user stories is according to the INVEST principles (Independent, Negotiable, Valuable, Estimatable, Small, Testable). For 'estimable' and 'small', each write a user story that is generally sensible in the Kiosk scenario, but violates (only) that principle. Afterward briefly explain why this user story violates the principle and provide an improved version of the same user story. Use the common *"As a [role], I want [function], so that [value]"* template.

i. User story violating (only) the *'estimable'* principle:

Briefly explain why it violates the principle:

Fixed version of the same user story:

ii. User story violating (only) the *'small'* principle:

Briefly explain why it violates the principle:

Fixed version of the same user story:

Question 1 continues. . .

(b) One reason customers like your app is that it is considered highly *usable.* As you expand, your management is concerned about maintaining this important quality as a market differentiator.

i. (4 points) (*1 NN*) What are two ways you might measure usability, and what are their tradeoffs?

ii. (4 points) (*1 NN*) Your colleague argues that usability cannot be measured and must instead by left to the intuition of the app designer. They say "all those metrics don't really capture usability, they're pointless." Do you agree or disagree? Why or why not?

Writing below this line is permitted but discouraged.

Question 2: Process and Metrics (14 points)

(a) (4 points) *(1 NN)* We have discussed two major case studies—Boeing and healthcare.gov—where process and culture concerns led to software failures. Briefly describe one of those issues for one of those case studies.

(b) (6 points) *(2 NN)* What is one process intervention you could implement in your company to avoid the same kind of failure?

(c) (4 points) *(1 NN)* The data scientist on your team thinks that software is always too buggy and suggests that the company start incentivizing high-quality code by looking at bad reviews of the apps and count who was responsible for the code that caused the problem the user complained about. The developers with the buggiest code should be then mentored better or sent to additional training courses. Do you think this is a good idea? Briefly explain why or why not.

Question 3: Architecture (22 points)

Your app is currently customized for Pittsburgh, interfacing with PAT data sources to access transit routes, locations, and real-time data, as well as PAT payment kiosks and on-vehicle services. You host the server-side computations on hardware in-house. Now, your company is planning out how best to architect and deploy the transit app to cities beyond Pittsburgh. You consider the following options: (There may be other options; stick with these for the purposes of this question.)

- Option 1: Stick with a single app, and your current in-house deployment. Customers select the city of interest from within the app.
- Option 2: Generalize your current framework such that it is more easily configurable to new cities; transition your hosting/deployment to a cloud-based service.

(a) (9 points) *(3 NN)* Architectural reasoning often involves *tradeoffs*, often with respect to quality attributes or non-functional properties. Identify three important quality attributes that might be affected, and describe (briefly) how each option affects it.

Quality	Effect of Option 1	Effect of Option 2

Question 3 continues…

(b) Your team has also been discussing the potential use of microservices in the overall architecture.

i. (3 points) *(1 NN)* List at least 3 services which would be reasonable for this project (descriptive names are sufficient):

ii. (6 points) *(2 NN)* Your existing application was developed monolithically. How would you transition to microservices? Do you recommend implementing new features as microservices and slowly transitioning, or re-writing the existing application from scratch right now? Give one argument in favor of each position and make a recommendation.

Argument for incremental transition:

Argument for full rewrite:

Recommendation (with brief justification in the context of the scenario):

(c) (4 points) *(1 NN)* Design documents consider goals, but also non-goals. As you write a design doc for potentially moving to microservices, give one goal, and one non-goal you would include in your design doc.

Question 4: Machine Learning and Ethics (18 points)

Your CEO decides that she wants your product to have "Artificial Intelligence". After several rounds of discussions, it is decided that you should use Machine Learning to analyze the data you collect, and sell a consulting service to cities, to help them cut costs by finding areas where there are bus routes that are not well subscribed, and then reducing or ending that route.

You decide to start using Machine Learning to predict how many riders will be on the bus at any given time. You will be working closely with a new Machine Learning expert that your company has hired. She will be responsible for your choosing and tuning the model. They are responsible for finding which features the model can be trained on. You are tasked with collecting the data that the model will use. You should choose data that you are already collecting, or can easily collect.

(a) (6 points) *(2 NN)* What features would you propose to use for the Machine Learning component?

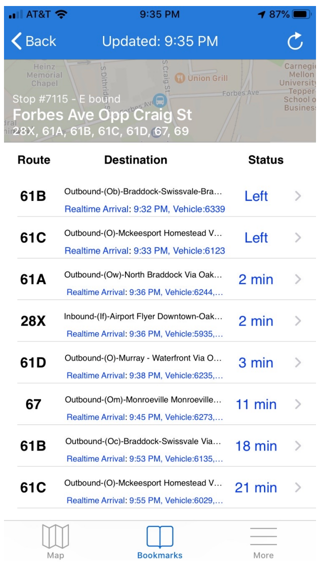
(b) (4 points) *(1 NN)* Are there any features that you would specifically choose to not use?

(c) (8 points) *(2 NN)* Before you begin to develop your feature, you decide to evaluate the ethical implications of this new feature. You are tasked with raising any ethical issues, and proposing ways to mitigate potential harm. Do you have any ethical concerns with how this new feature may be used? Why or why not?

Question 4 continues…

A Appendix: Scenario description

The following fictional scenario will be the basis of most questions within this exam. You may detach this page from the exam.


AT&T 9:35 PM 87%

Back Updated: 9:35 PM

Stop #7115 - E bound

Forbes Ave Opp Craig St

28X, 61A, 61B, 61C, 61D, 67, 69

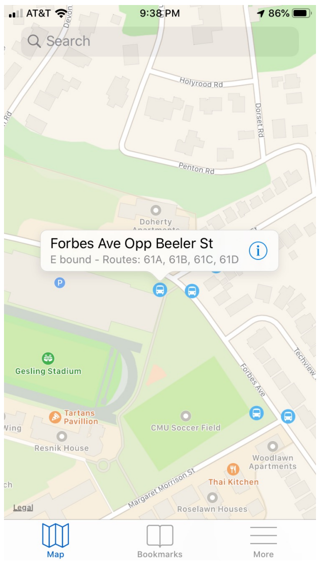
Route	Destination	Status
61B	Outbound-(Ob)-Braddock-Swissvale-Bra… Realtime Arrival: 9:32 PM, Vehicle:6339	Left
61C	Outbound-(O)-Mckeesport Homestead V… Realtime Arrival: 9:33 PM, Vehicle:6123	Left
61A	Outbound-(Ow)-North Braddock Via Oak… Realtime Arrival: 9:36 PM, Vehicle:6244,…	2 min
28X	Inbound-(If)-Airport Flyer Downtown-Oak… Realtime Arrival: 9:36 PM, Vehicle:5935,…	2 min
61D	Outbound-(O)-Murray - Waterfront Via O… Realtime Arrival: 9:38 PM, Vehicle:6235,…	3 min
67	Outbound-(Om)-Monroeville Monroeville… Realtime Arrival: 9:45 PM, Vehicle:6273,…	11 min
61B	Outbound-(Oc)-Braddock-Swissvale Via… Realtime Arrival: 9:53 PM, Vehicle:6135,…	18 min
61C	Outbound-(O)-Mckeesport Homestead V… Realtime Arrival: 9:55 PM, Vehicle:6029,…	21 min

The product. You work for BusMagic, a small, fictional technology company that specializes in creating custom transit apps for cities across the United States. These apps are intended to allow individuals to learn about and use local transit options by (a) searching a map for directions for transit routes between destinations, and (b) selecting stops on the map and query them for information upcoming buses or trains.[1]

The app accomplishes this by interfacing with and querying information from a municipality's existing transit infrastructure. Additionally, for cities whose transit systems support it, the app offers the ability for customers to pay fares using their phone (using the Google Pay or Apple Wallet functionality).

The product. Your company started in Pittsburgh, but other cities have taken note of the app and are expressing interest in having your team support their infrastructure as well. Your company presently makes money by contracting with the city directly. However, you have also been discussing the possibility of additionally serving (unobtrusive!) ads to users, and potentially offering an ad-removal option to be paid directly by the consumer.

The team. You are on one of several small teams within the company focusing on backend concerns; others are in charge of testing/QA, and frontend UI. You work with two senior members with considerable mobile development experience, as well as three additional, more junior employees, and a data scientist.



[1]These screenshots are of Pittsburgh's BusGazer app, but they illustrate the idea.