CS312 Spring 2024 – Midterm 1

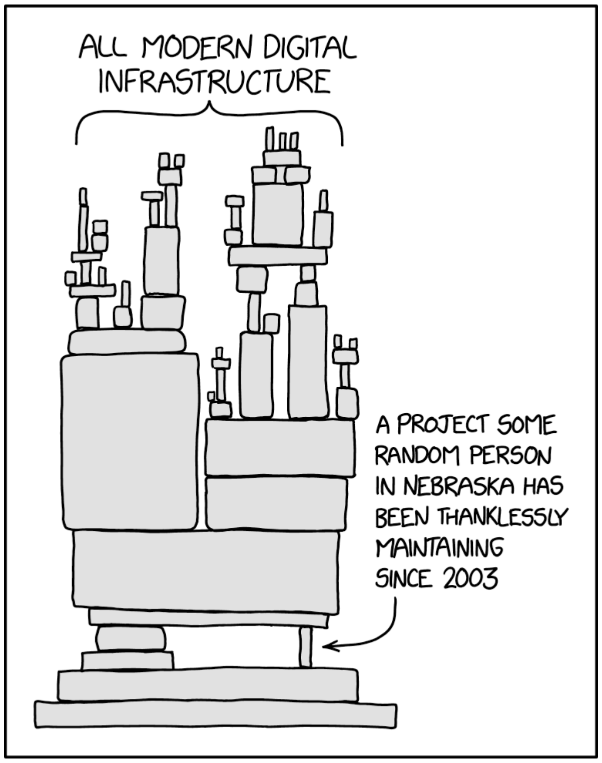
Name: ______________________________

Date: ____________________ **Start time:** ________________ **End time:** ________________

Honor Code:

Signature: ______________________________

This exam is open course web page, open Ed, open notes, open slides, open your assignment solutions and open calculator, but closed everything else (e.g., consulting with others, searching online, using generative AI are not permitted). **You have 2 hours in a single sitting to complete the exam.** Read the problem descriptions carefully and write your answers clearly and *legibly* in the space provided. Circle or otherwise indicate your answer if it might not be easily identified. You may use extra sheets of paper, stapled to your exam, if you need more room, as long as the problem number is clearly labeled and your name is on the paper. If you attached extra sheets indicate on your main exam paper to look for the extra sheets for that problem.


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Learning Target	Assessment
1	
2	
3	
4	
5	
6	
7	
8	

Question 1. User stories

You are developing a web application for managing a library. When interviewing stakeholders, multiple respondents described wanting to reserve books online for subsequent pickup. Write two I.N.V.E.S.T. user stories for this feature, one from the perspective of a library patron, the other from the perspective of a library staff member. Your user stories will be evaluated on format and quality.

(a) Library patron:

(b) Library staff member:

Question 2. Javascript

Assume the function `any_n(promises, n)` returns a promise that resolves when all, or at least `n`, of the promises in the array `promises` have resolved, whichever happens first. The promise returned by `any_n` will resolve with an array of the fulfilled values of `promises`. The function `wait(sec)` returns a promise that resolves after `sec` have elapsed.

```
function do(time) {
  return wait(time).then(() => {
    console.log(time)
  });
}

do(3);
any_n([do(1),do(2),do(4)], 2).then(()=>{
  console.log(5)
});
```

```
async function do(time) {
  await wait(time);
  console.log(time);
}

await do(3)
await any_n([do(1),do(2),do(4)], 2);
console.log(5);
```

Consider the two code snippets above. Write the expected output for left-side code below on the left. If the right-side code produces the same result indicate below, otherwise provide the expected output below on the right.

◯ Both snippets produce the same output

Question 3. Testing

You are developing a React component named `PasswordGenerator` for generating a random password. The component has a select box for specifying common allowed character sets, e.g., "letters and numbers", a numeric input for specifying the length, a "Create" button to create a new password, and a text field to display the generated password. Using the skeleton below, implement **pseudo-code** for a F.I.R.S.T. unit test to verify that each time the user clicks the "Create" button a new correctly formatted password is generated. You do **not** need to provide executable Javascript, instead describe the steps of your test as pseudo-code. For example, one of the steps in your pseudo-code might be:

Assert mock function was not called

You may or may not need all of the functions below. You only need to include pseudo-code in bodies of the functions relevant to your answer.

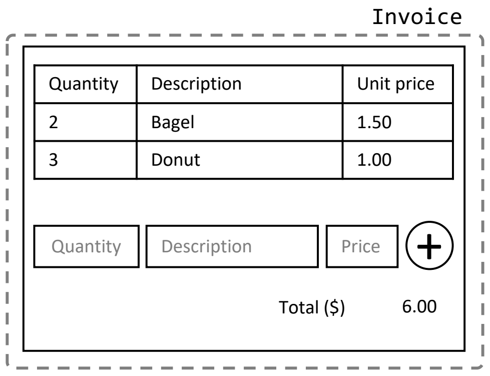
```
describe("Password generator", () => {
  beforeEach(() => {









  });
  afterEach(() => {









  });
  test("Generates a new correctly formatted password (letters/numbers) on each click", () => {
















  });
});
```

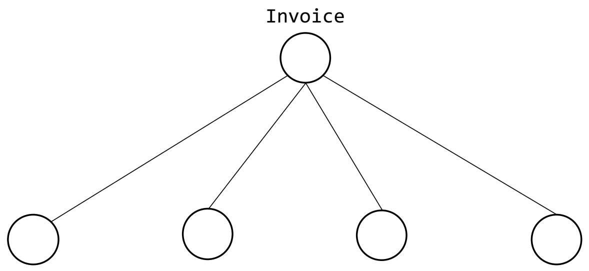
Question 4. Scenarios

In your application, the user profile page has a password field with an associated “Update” button to update the user’s password. If the user updates their password (to a new password), the user is logged out, redirected to the login page, and a confirmation message is shown in a “flash” (i.e., a message that is shown for a short period of time as a banner at the top of the page). Write a Gherkin-style test scenario for updating a password. You do not need to provide the implementation details of the tests, just describe the scenario for the test.

Question 5. React

You are implementing the invoice creator shown below with React. Entering a integer quantity, string description and decimal unit price and clicking the "+" should add an entry to the invoice and update the total at the bottom (as the sum of the quantity times the unit price for all entries). Outline and label the wireframe (below, left) with a possible set of components. Label the tree (below, right) with components to show the hierarchy. Label the tree nodes with state implemented in that component and label the tree edges with props passed to each component (similar to the figure in programming assignment 2). Repeated components can be labeled once in tree. The top-level component `Invoice` is labeled for you. Any implementation reflecting good React practices will be accepted. You may not need all the nodes in the tree or may need to add nodes depending on your design; cross out any unused nodes. Your component, state and prop names should be sufficiently descriptive that their role is clear.





Question 6. REST

For each of the following pages in a NextJS-based online store, provide an appropriate RESTful front-end (browser) URL for that page and, where relevant, an appropriate RESTful server API endpoint (HTTP verb and URL) that component would interact with. An example is provided below.

Page	Page URL	API HTTP verb and URL
Add a new article to Simplepedia	**/edit**	**POST /api/articles**
Change the price for a product	______	______
View products sorted in ascending order of price	______	______
View a specific user's past orders	______	______

Question 7. Data modeling

Assume you are developing a web application for supporting student clubs at a college, e.g., membership lists, calendars, announcements, etc. You will be using a relational database to store the data for this application.

(a) Identify the *minimum* set of models you would define in your server backend to implement the following user story:

As a student, I want to view a consolidated a list of announcements for all the clubs I am a member of, so that I can stay informed about all club activities.

(b) Which of the following best describe the relations between the following pairs of entities. Select one answer for each pair, then briefly explain your answers.

Student and Club	Club and Announcement
○ One-to-One	○ One-to-One
○ One-to-Many	○ One-to-Many
○ Many-to-Many	○ Many-to-Many
○ No relation	○ No relation

(c) In a normalized schema designed for a relational database (RDBMS), what schema would be needed to support club membership. Assume club members can have different roles, e.g., "member", "president", etc. You do not need to provide SQL, just the attributes, their types, the primary key, and any foreign key constraints.

Question 8. Development processes

For each of the following, indicate whether the action would be consistent with the best practices for software development as described in class or not consistent. Here "consistent" is defined as consistent with good development practices generally, not that it was required as part of our class. Briefly explain your answer.

(a) Each team member uses a separate "personal" branch throughout the sprint, e.g., `mlinderman_sprint1`, to implement their tasks before merging with `main` at end of the sprint.

◯ Consistent ◯ Not consistent

(b) Assign a responsible developer for all the tasks in the sprint backlog during the sprint planning meeting.

◯ Consistent ◯ Not consistent

(c) Update the `main` branch and rebase a newly created feature branch before pushing that feature branch to GitHub for the first time.

◯ Consistent ◯ Not consistent