

Case Studies in Software Design - Examination, April 2022

Section A: Unseen Questions

1a

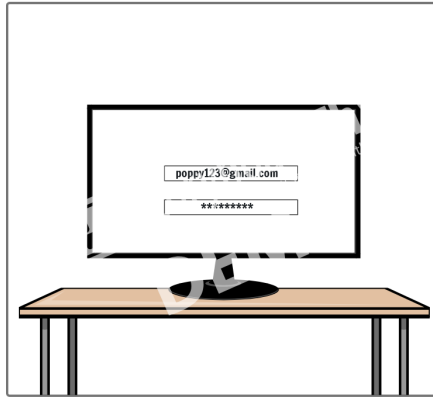
Persona 1 Poppy is a 23 year old university student studying at Manchester university for a Masters in Sociology. Despite her busy workload, she loves travelling around the country to visit different cities and rural areas, and looks for new active experiences like rock climbing or trampolining which keeps her fit and healthy. Her Fiat 500 has been a trustworthy companion to get her around the country. She is tech-savvy, especially with her iPhone, using it to find deals and book these experiences to help balance costs on a university student budget.

Scenario 1a Poppy is driving in her Fiat to Sheffield to visit some friends for the day who attend university there, aiming to watch a film and raid as many city-center vintage stores as possible. Before starting her journey across the Snake Pass, she used her desktop PC to book a parking spot near the cinema via Paypal, as well as her cinema ticket which she saved to her smartphone. Upon arrival, she slots her car into the allocated bay and departs for the Moor to meet her friends.

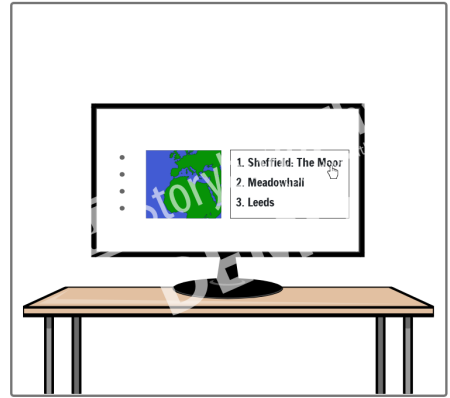
Storyboard 1a



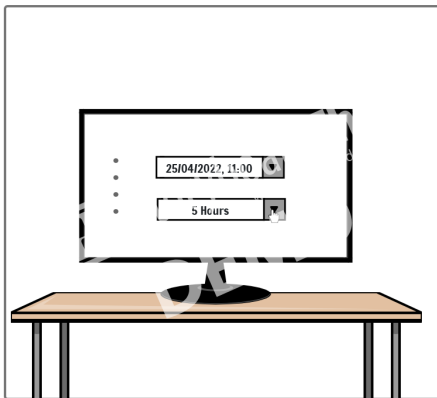
Poppy has finished getting ready for day and is booking her parking in Sheffield for the day.



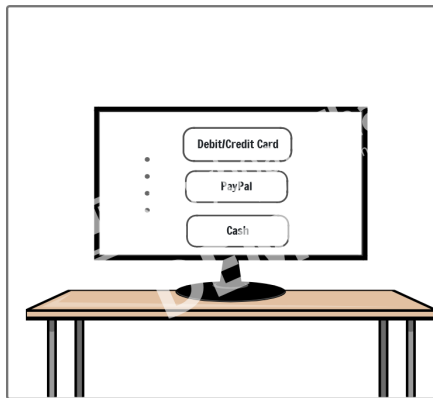
She opens the website on her PC and logs in with her email and password.



Using the map, she finds the nearest car park to the cinema.



After discussing the day plan with her friends, she estimates how long she will be in Sheffield and enters the required amount of hours, plus the start time.



She chooses to pay via PayPal.



Once the payment goes through, a confirmation screen is shown.

Create your own at Storyboard That

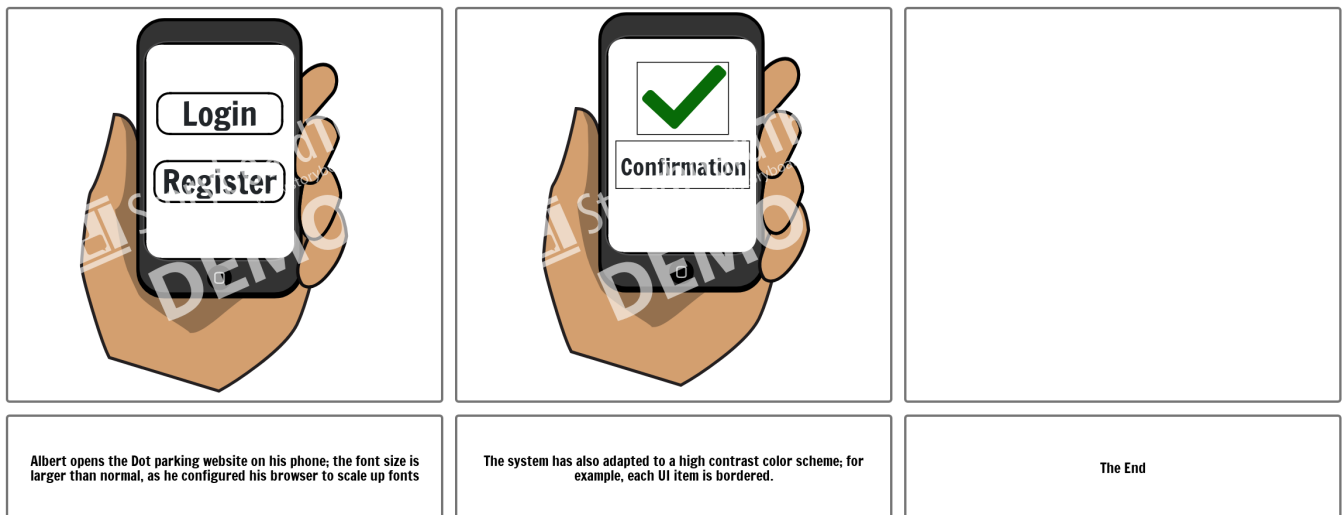
Scenario 1b It has been almost 5 hours since Poppy parked up in Sheffield and her friends have encouraged her to extend her visit. They are all sat in a cafe with free WiFi, of which Poppy takes advantage. After joining the network, she she logs into the parking system on her phone , finds her current parking allocation and adds 2 more hours. She wants to get back to her friends, so she neglects to pay in advance as she knows she has the cash to pay later.

Storyboard 1b (Ran out of time)

Persona 2 Albert is a 62-year-old accountant who works from home, occasionally commuting into the London office in his Jaguar XK. Considering his age and life-long desk job, his physical health is very good thanks to his morning runs, however he does suffer with an eye condition which makes it difficult to read small print and digital screens. As a hobby, he tinkers with miniature circuit boards and arduinos to create useful gadgets to use around the house.

Scenario 2 It's 06:30 on the last Friday of the month and his accountancy firm is having an office get-together after work. Uncharacteristically, Albert has decided to pop along so he plans to work in the office today. He does not have a designated parking space due to his preference for WFH, so he decides to book a parking space online. When he opens the DoT parking website on his phone, its font size is scaled up by the browser and defaults to a high contrast color scheme which he has configured in his phone's settings. From here, he has no issues reading the website's text and books the parking space he needs.

Storyboard 2



Create your own at Storyboard That

1b

Booking a parking spot is a very structured process. A user selects a parking location, then a time, then the amount of hours, then a payment option; upon completing payment, the process is finished and the user can park their vehicle. The UI should use a process-oriented pattern such as the Steps Left pattern. It displays the steps in a process (usually) as a timeline. The state of each step is indicated with an icon, color, and optionally text; e.g., a 'Completed' step would be green and its timeline node set to a tick.

1c

Resetting a password is required functionality of a software system using password-based authentication; users can simply forget passwords, or they may wish to change it periodically, or an account with the same password on another system may have been compromised. This applies to any user in the system such as employees, registered user and subscription accounts. In any case, this task would follow this process:

1. Open the DoT parking website
2. Navigate to the login page
3. Click the 'Reset Password' link
4. Confirm your known authentication credentials; i.e., email, username, full name, etc.
5. Follow the link sent your email and submit your new password

Section B: Seen Questions

Designing Software Systems

One persona we created is Hannah, an elderly woman currently working as a toll operator. Her scenario establishes how an employee uses the system, plus how she configures the system to meet her accessibility requirements and its alignment with the previous system. Since the system is used exclusively in the manner described, all of its accessibility requirements are covered in Hannah's detailed scenario.

Dave, a regular tourist in Norway, is another persona we created. He has two scenarios explaining how a tourist uses the system: where the entered and exited, how a postal bill is received, how they change its language, plus the handling of multi-national registration plate and the manual fallback which utilises toll

operators. In addition to the residents of Norway who will not use an RFID tag, the high number of tourists also make these detailed scenarios highly relevant important to ensure the usability of the system for all highway users across Europe.

Implementation of the System

Our implementation is composed of an API serving a web app. The API is horizontally architected with a presentation, service, and data layer.

The Database component, specified in the Architecture Overview, is implemented by the data layer using the Repository pattern. The `IBaseRepository` interface defines basic CRUD functionality, taking a generic argument to specify the domain entity (e.g., `Bill`). This base functionality is implemented in the `BaseRepository` class. Each domain entity defines a specific interface and concrete class, deriving from the bases (e.g., `IBillRepository` and `BillRepository`). By abstracting the database provider behind repositories, it can be configured and/or swapped without affecting the service layer above which improves the system's maintainability. Entity-specific repositories allow for organised re-use of common queries and mutations, and all implementations can be tested.

Another component in the API is the External Payment Provider which handles payments through the online portal. A real-world provider like Stripe would be used in production, but an external system is not suitable for development. As well as creating additional overhead, test accounts would not be usable and real payment information is inappropriate and unsafe to use for local development, staging, etc. Since the API follows the Dependency Injection pattern, services are defined with an interface (e.g., `IExternalPaymentProviderService`) and their appropriate implementation is determined at runtime by the provider via Inversion of Control (IoC). In development, the `TestExternalPaymentProviderService` is provided.