Software Design & Architecture

Human-Perspective Requirements

Agenda

- User scenario
- Use case
- Human values in software engineering

Winter 2026. Instructor: Pengyu Nie.
Acknowledgements: slides adapted from previous versions by Mei Nagappan and Shane McIntosh, which are adapted from previous versions by Zhen Ming Jiang, Ahmed E. Hassan, Reid Holmes.

User Scenario

- A story describing how a user may use your system:
  - goal
  - context and constraints
  - success criteria
- Discover requirements, assumptions, and risks

User Scenario Example

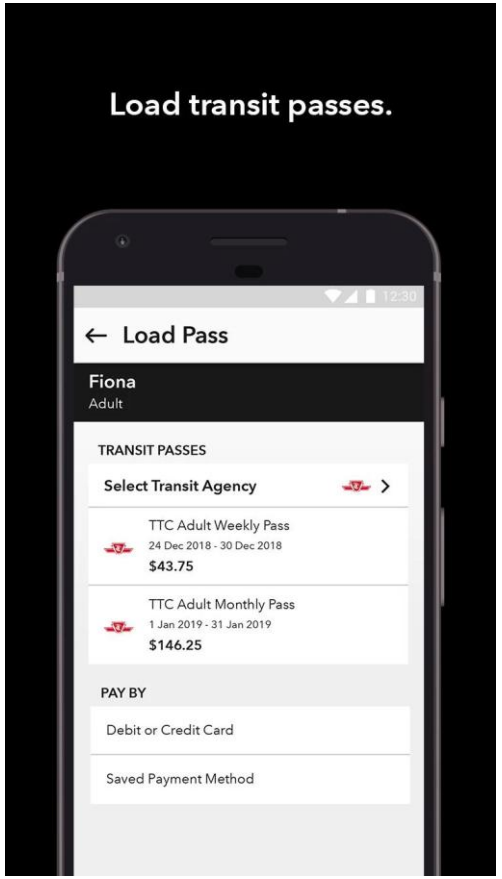
Transit ticketing app – Buy and Board

Riley is heading home after work and reaches the bus stop just as the next bus is about to arrive. They usually use a physical transit card, but they left it at home today.

Riley opens the transit ticketing app to buy a single-ride fare. Riley wants to complete the purchase quickly and be able to show proof of payment immediately when boarding.

The app guides Riley through selecting a single-ride ticket and paying with a saved payment method.

After purchase, the ticket appears in the wallet with clear status information and a prominent way to display it for inspection.



Developing User Scenarios – questions to ask

- Gather inputs from users and stakeholders
- Identify specific goals and triggers
  - What are you trying to achieve?
  - What starts the story?
- Capture context and constraints
  - Who/Where/When
  - What factors may make it hard?
- Define success outcomes
  - What would count as success?
  - What would count as failure?
  - If the system fails, what do you do instead?

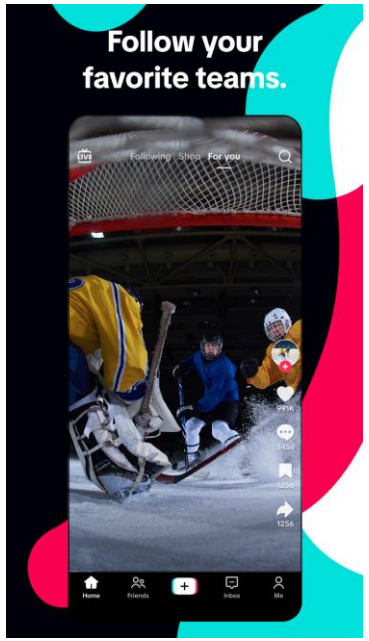
Developing User Scenarios – strategies

- Be specific, not generic
- Write for a named persona, not “the user”
- Keep it short, just enough flow to be concrete, no detailed steps
- Avoid mixing architecture/design decisions
- End with an observable success criterion

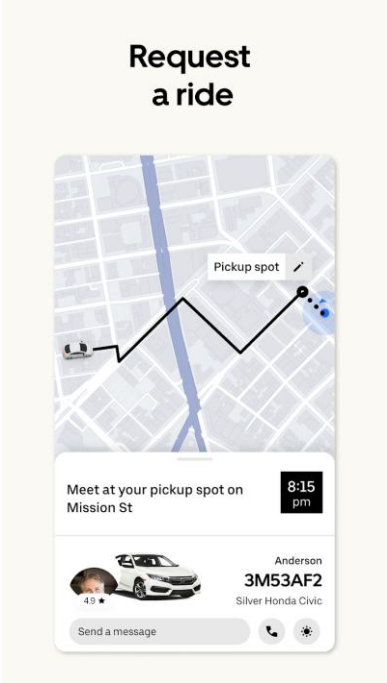
Exercise: Drafting a User Scenario

- Pick one of the app below
- Discuss and come up with a user scenario for the app

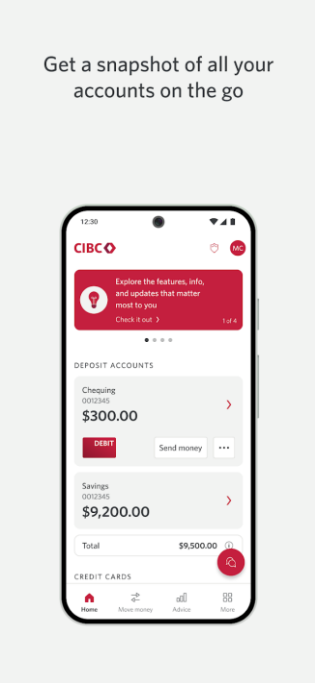
short video app



ride sharing app



online banking app



... or your project idea

Usage Scenario -> Use Case

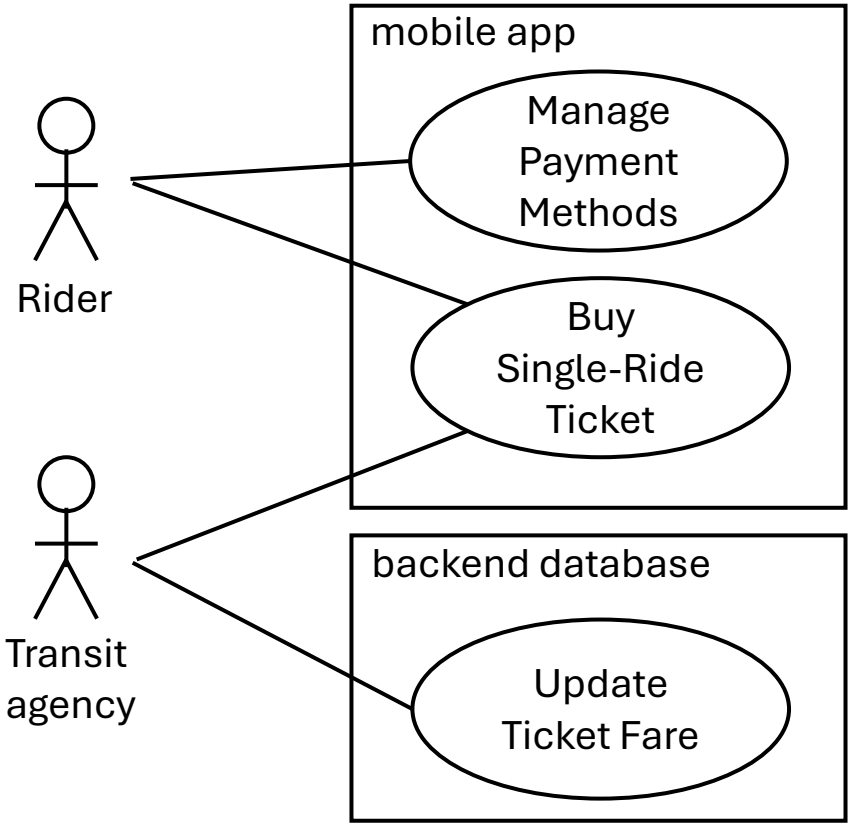
- User scenarios are great for discovery
  - + capture goals, context and constraints, success criteria
  - + expose assumptions and risks
  - - not specification
  - - not complete
- Use case: modeling requirements from human perspective
  - + specify actors, main success scenario, alternative flows …
  - + contract between stakeholders and developers
  - + enable testing

Use Case Diagram

- Use case diagram (in UML): overview of all use cases in the system

Actor

- a user in a particular role
- human / organization / external system



Use Case

- a task with business value that an actor needs to perform with the help of the system
- main success scenario + failure alternative scenarios

Use Case Description Template

ID	
Name	short phrase, usually verb + noun
Actors and interests	the actors/roles involved, and each one's goals
Trigger	an event that starts the use case
Preconditions	system conditions for the use case to normally start
Postconditions	system conditions after the use case is finished success guarantee / minimal guarantee
Main success scenario	the steps in a normal successful user scenario
Extensions	alternative flows and exceptions
Special requirements	related non-functional requirements

Use Case Description Example

ID	UC1
Name	Buy Single-Ride Ticket
Actors and interests	* Rider (primary): fast purchase, correct fare, ticket available for inspection. * Transit agency: correct fare collection, reduced support burden.
Trigger	Rider chooses to buy a ticket in the app
Preconditions	Rider is logged in A payment method or stored balance is available
Postconditions	* Success guarantee: a valid single-ride ticket is issued to the rider and is visible in the app wallet; payment is completed and recorded. * Minimal guarantee: the rider receives an actionable error message, and no unaccounted charge is captured.

Use Case Description Example (cont.)

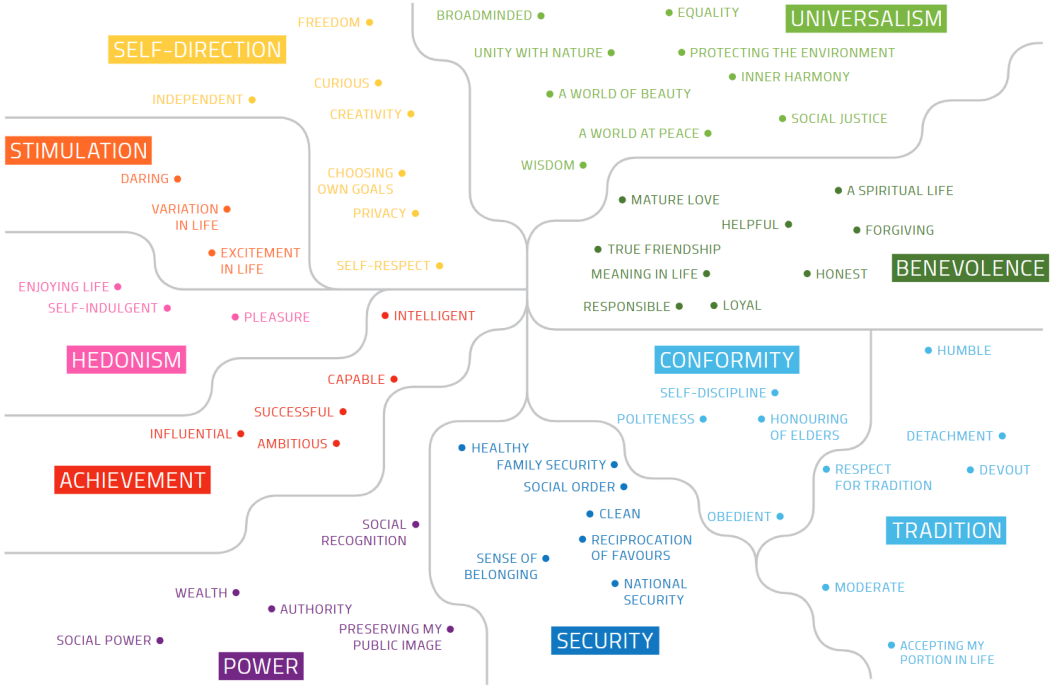
Main success scenario	1. Rider selects “Buy Ticket”. 2. App shows available fare products. 3. Rider selects “Single Ride” and confirms purchase details. 4. Rider selects a payment method and authorizes payment. 5. App confirms success and navigates to the added ticket in Wallet.
Extensions	1a. Rider is not logged in. 1a1. App prompts for login. 1a2. Rider logs in successfully; resume step 2. 1a3. If login fails, end. 4a. Network times out. 4a1. App informs the rider and offers Retry or Cancel. 4a2. If Retry, repeat step 4. 4a3. If Cancel, end.

Use Case Description Example (cont.)

Extensions	4b. Payment declined. 4b1. App informs the rider and offers another payment method or Cancel. 4b2. If another payment method chosen, resume at step 4. 4b3. If Cancel, end. 5a. Ticket creation fails after payment authorization. 5a1. App informs the rider and initiates payment reversal request. 5a2. App records the incident and directs the rider to support; end.
Special requirements	1. A rider who is already logged in and has a saved payment method shall be able to purchase a single-ride ticket from the home screen in no more than 3 taps. 2. When retrying, the app shall preserve the rider's existing selections so they don't need to redo earlier steps.

Human Values in Software Engineering

- Software are built for humans
  - user scenarios & use cases:
    goals that humans want to achieve with software
  - good enough to support these goals?
- People have (implicit) values
  - high-level things that people care about
  - priority varies by time, individual, culture, …
  - and conflicts exist



From talk by Jon Whittle
at SIGGRAPH Frontiers
on Human Values in Software
https://youtu.be/845fORAf0FQ?t=270

I recommend watching this talk offline

Potential Harms from Software

- Software products are built with a purpose in mind, but that purpose may harm human users by not respecting their values
- Harm can be caused:
  - by a feature with (unintended) consequences
  - due to the lack of a feature that a subset of users would deem necessary

Features with Consequences

- What might be some (unintended) consequences of the following features? How they might be avoided?
  - SelfieDrone: A drone system that can follow a user, take amazing selfies, and can even remove photobombers from your IG-worthy pictures
  - Messaging: An instant messaging platform like Discord or Slack, which allows any user to message anyone else on a platform
  - ResumeFilter: A system that automatically screens resumes to select qualified interviewees
  - InfiniteScrolling: The “infinite scrolling” feature built into most social media and shopping apps

Lacking of Features with Consequences

- What would be some key features of the below systems, such that if those features were missing, it may cause harm?
  - Ad Recommendation: A service that matches advertisements with users most likely to click-through on ads
  - Video Conference: A video conferencing system like zoom
  - Wearable Health: A wearable with a health app that monitors sleep times, heart beat rate, and step counter
  - News: News aggregator apps like Google News or Apple News

Recap

- User scenario
  - goal, context and constraints, success criteria
- Use case
  - diagram
  - itemized descriptions
- Human values
  - harm by features and lacking of features

- Reminder: P0 team formation due this Friday