THE UNIVERSITY OF MELBOURNE

IT Capstone Project

COMP 30022 2025

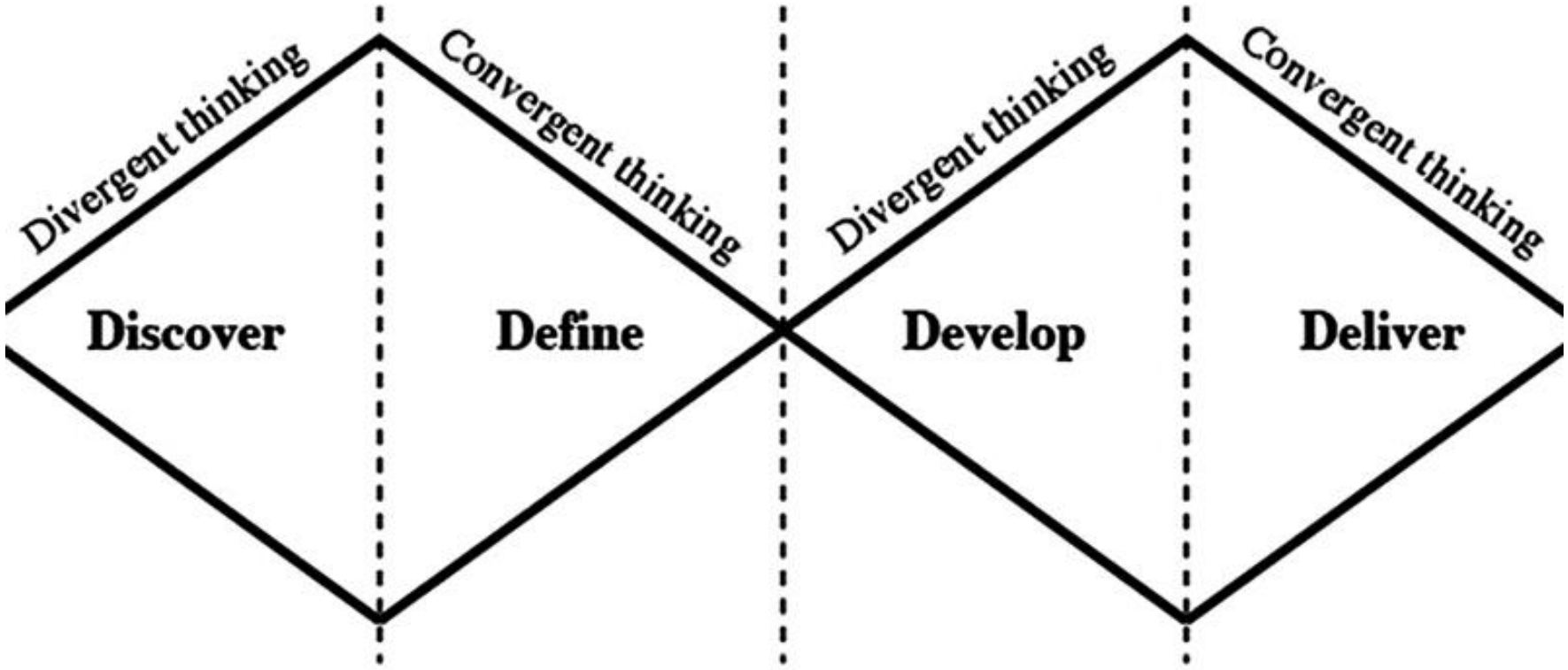
Week 4 - Design

THE UNIVERSITY OF MELBOURNE

Overview of Lecture

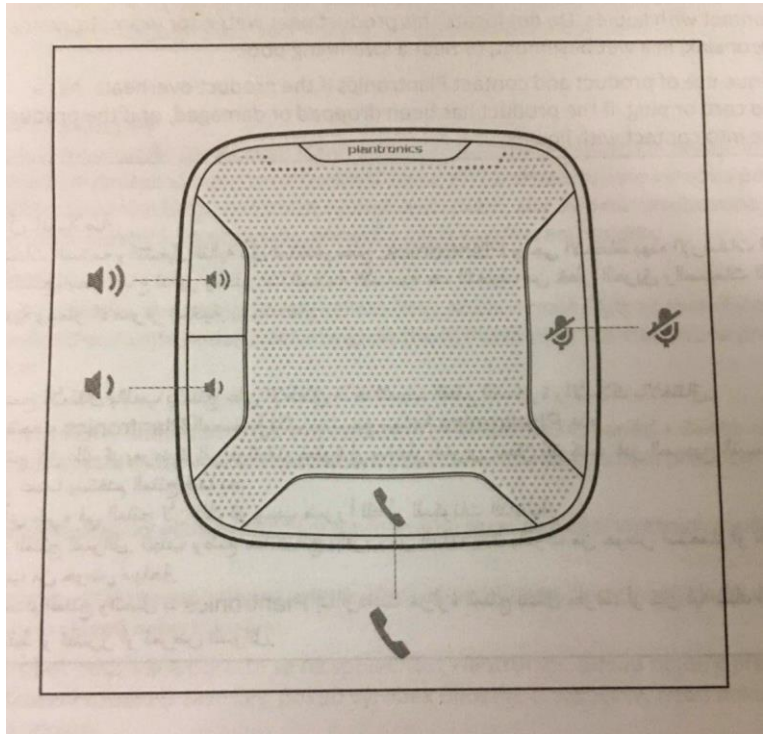
- Architecture Design
- 4+1 Architecture Model
- Front-end Design
- Low & high fidelity prototypes

Double Diamond Model (from design)



Design

- A communication exercise so people can understand your code (and thinking)
- No prescribed method
- UI design (client involvement will vary)
  - Plenty of templates: Figma, Canva, …
  - Wireframes, Low fidelity prototypes, …
- High level architecture
- Detailed architecture



THE UNIVERSITY OF MELBOURNE

High-level architecture

- For Web apps: A front-end, a back-end, and integration
- Front end concerns the user interface
- Back end concerns storing and retrieving information

THE UNIVERSITY OF MELBOURNE

4+1 Architecture Model

- Common Model for documenting software architecture
- Originally developed by Kruchten in 1995
- Defines a set of views, relevant to different stakeholders
- Makes it easier to understand a complex system

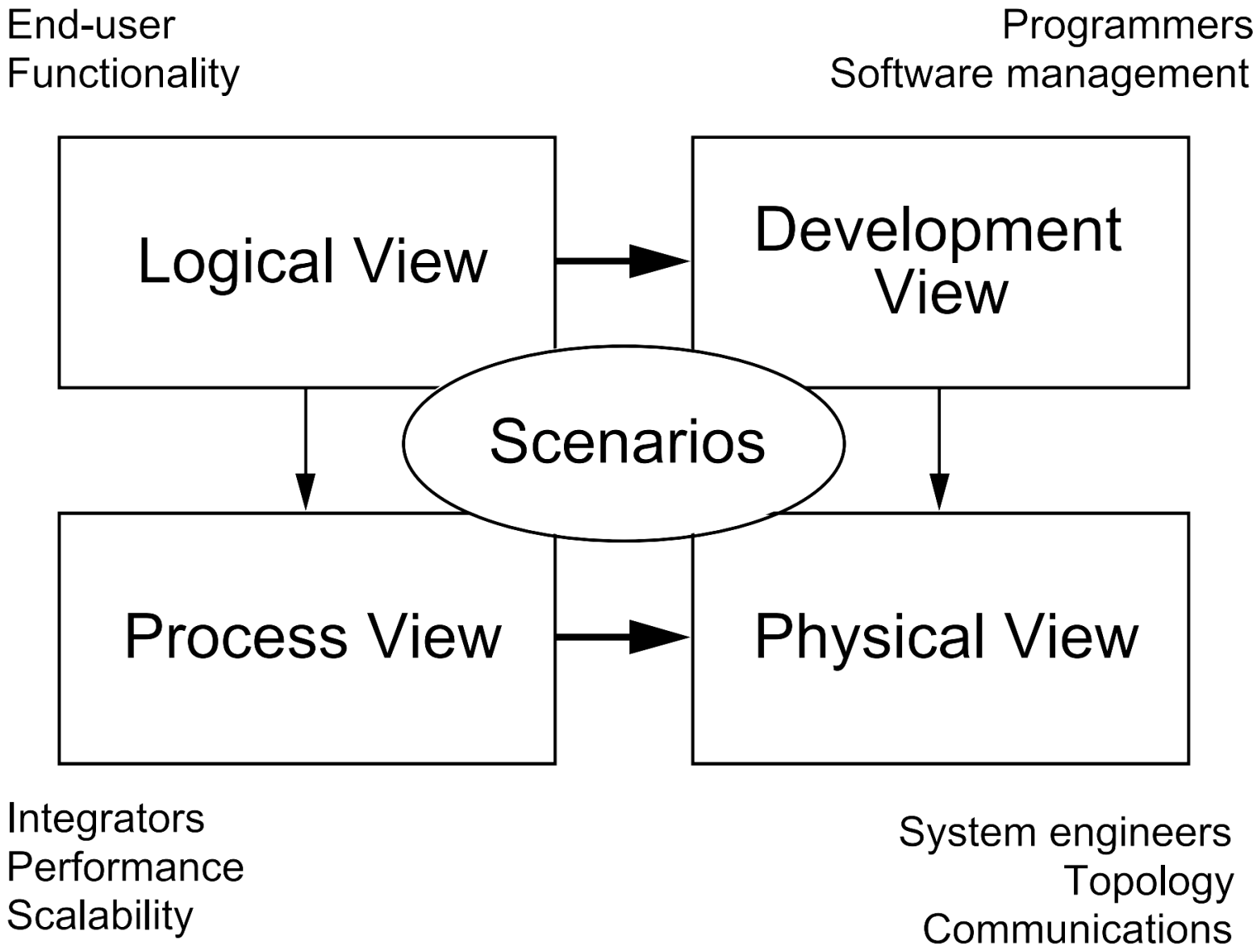


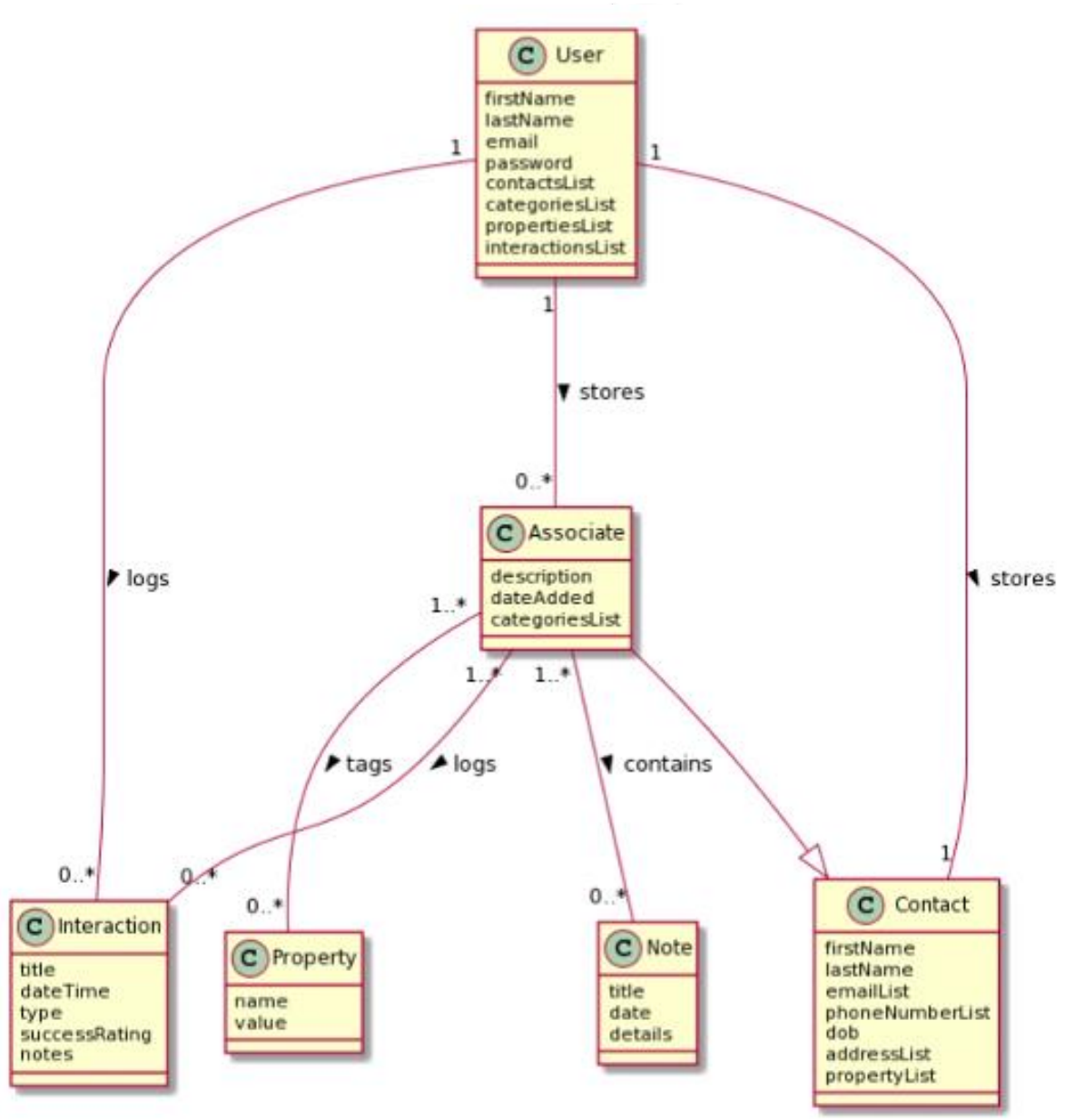
Figure 1 — The “4+1” view model

Logical View

- Describes the functional requirements of the system
- Shows the components of the system and their relationships
- Includes domain, class and database diagrams

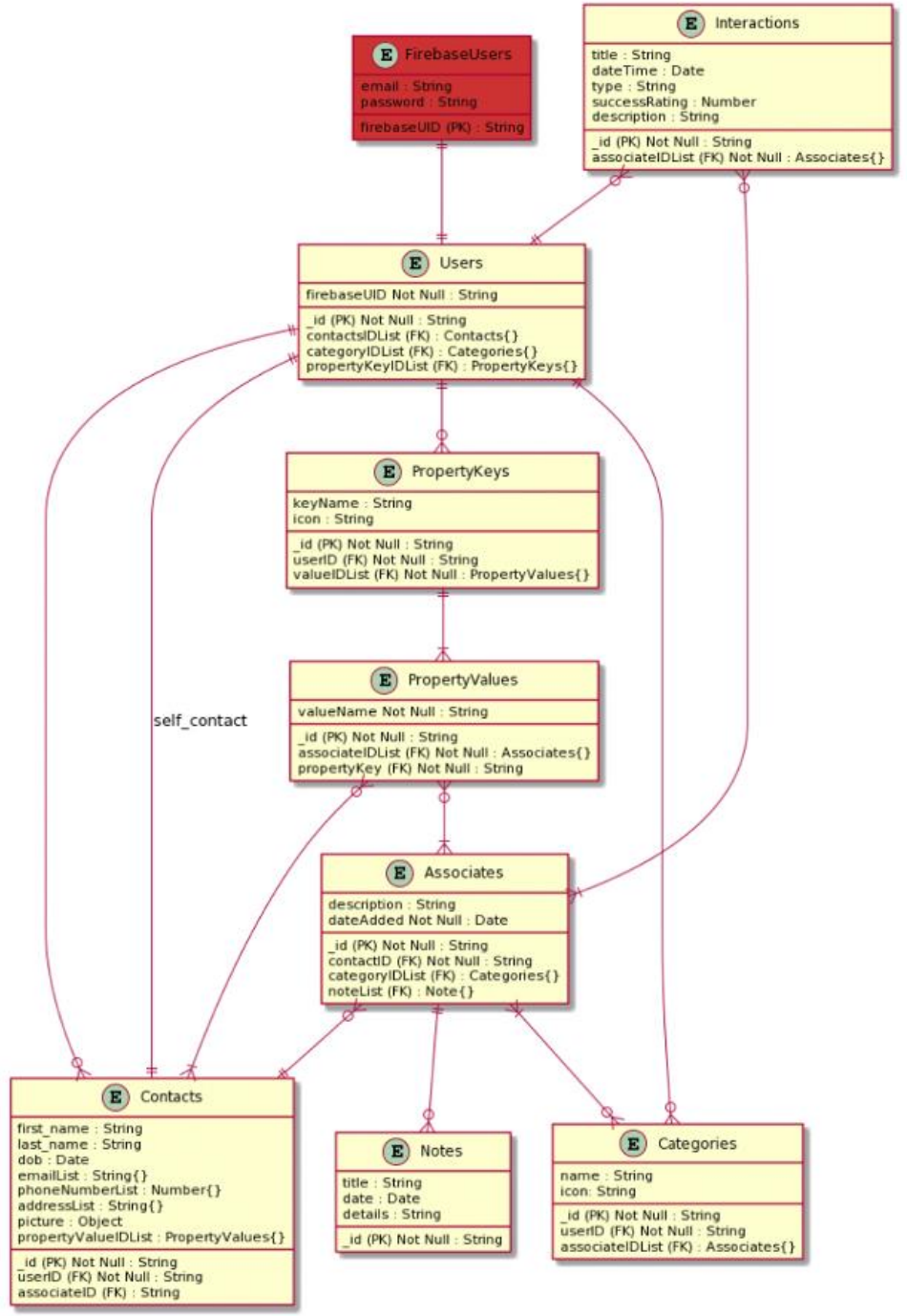
Domain Model

Domain Model Semantics	
Term	**Definition**
Category	Associates are categorised into concrete categories. The categories, 'Family Members', 'Friends' and 'Favourites' are added by default. More can be added by the user if they wish.
Property	Users have the option to create custom properties for any of their contacts. Examples of properties may be a favourite sport, food or city of residence.
Interaction List (Associate)	A log of interactions with a particular associate.
Interaction List (Customer)	All interactions of a user.



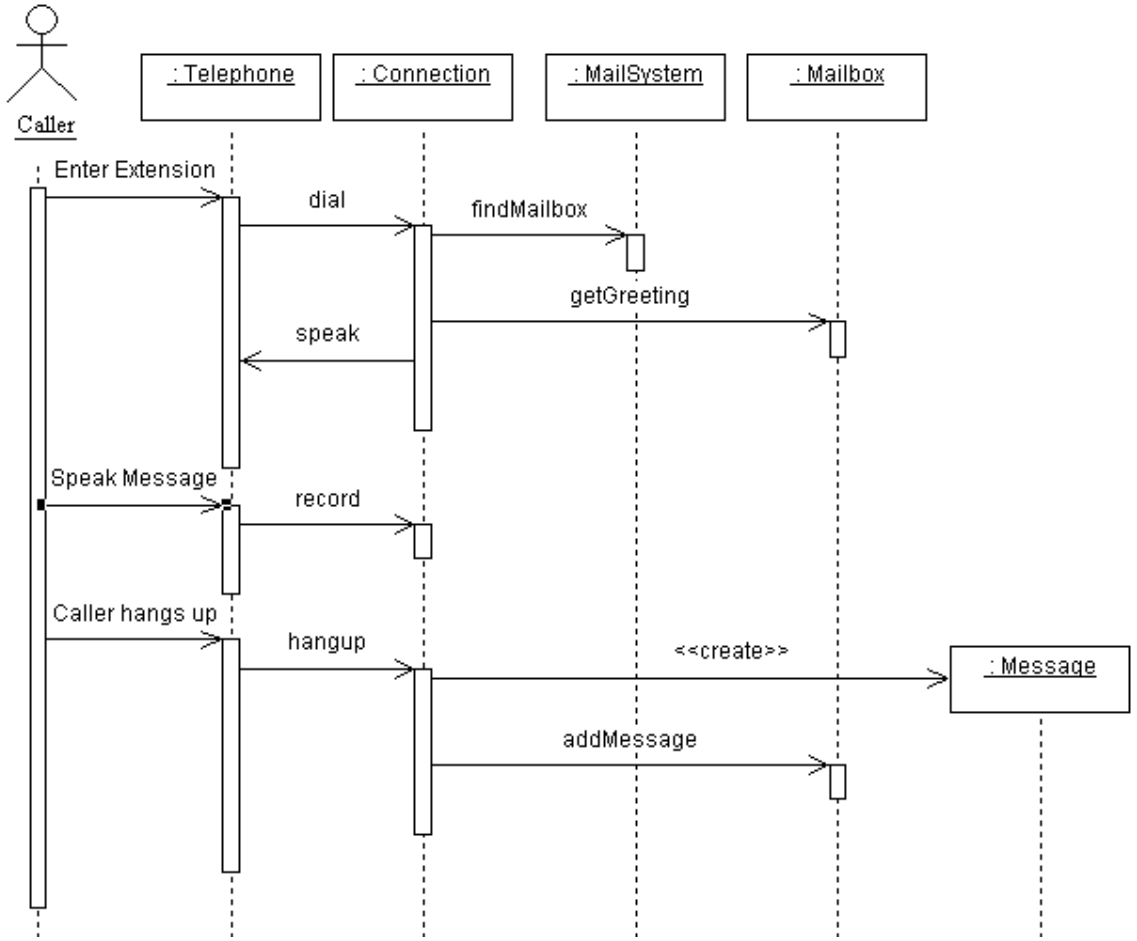
Database Model

- Provides a visualisation of database setup, e.g. with MongoDB
- Consider authentication, etc…



Process View

- Deals with dynamic aspects of the system
- Explains system processes and how they communicate
- Focuses on runtime behaviour of the system
- Includes sequence state diagrams



THE UNIVERSITY OF MELBOURNE

Development View

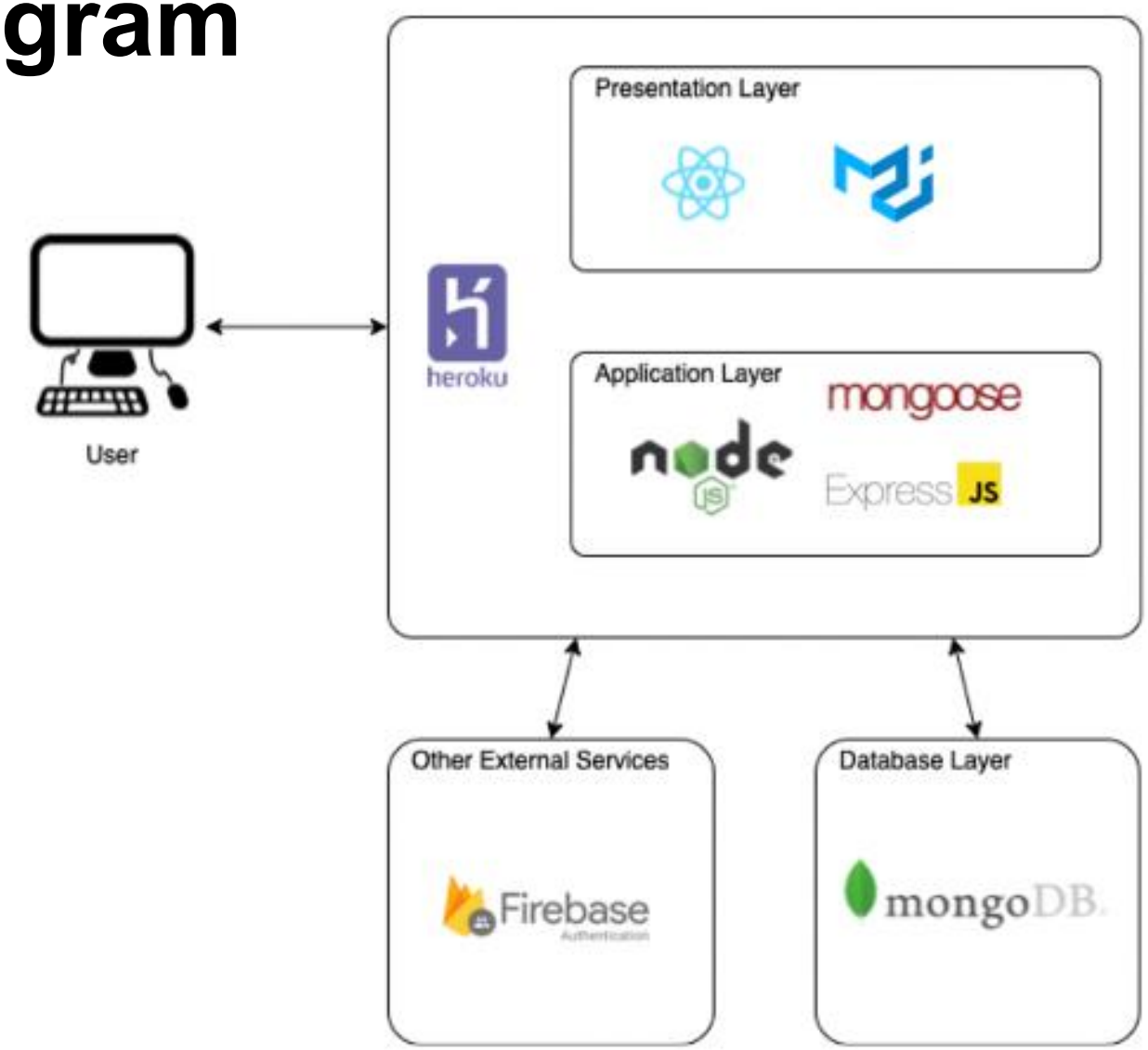
- Represented by package diagram
- Includes illustration from programmer's perspective
- Can describe architecture goals and constraints, system diagrams, API descriptions, etc…

Architecture Goals & Constraints

Requirement	What	Why	How
Authentication	System should verify a user is who they say they are	To protect sensitive data from unauthorised access	• Require users to log-in to correct account before providing sensitive information • Use Firebase Authentication
Confidentiality	System should ensure sensitive information is only accessible to those who are authenticated	To protect sensitive data from being utilised maliciously & maintain trust from users	• Encrypt data in transit (HTTPS) • Use secure methods of storing information in database
Data persistence	System should ensure data is saved and can be accessed later	To Prevent data provided from getting lost	• Use a database (e.g. MongoDB)
…	…	…	• …

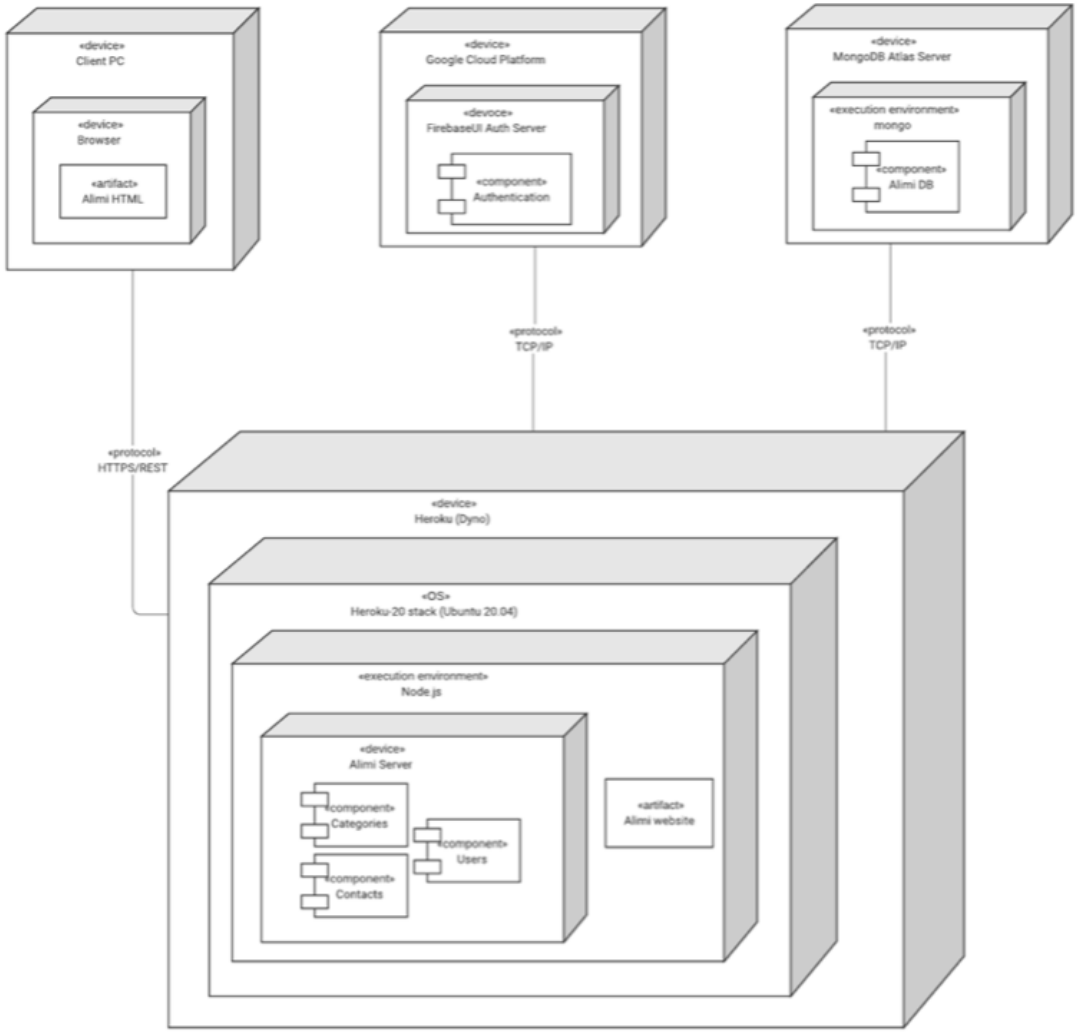
THE UNIVERSITY OF MELBOURNE

System Diagram

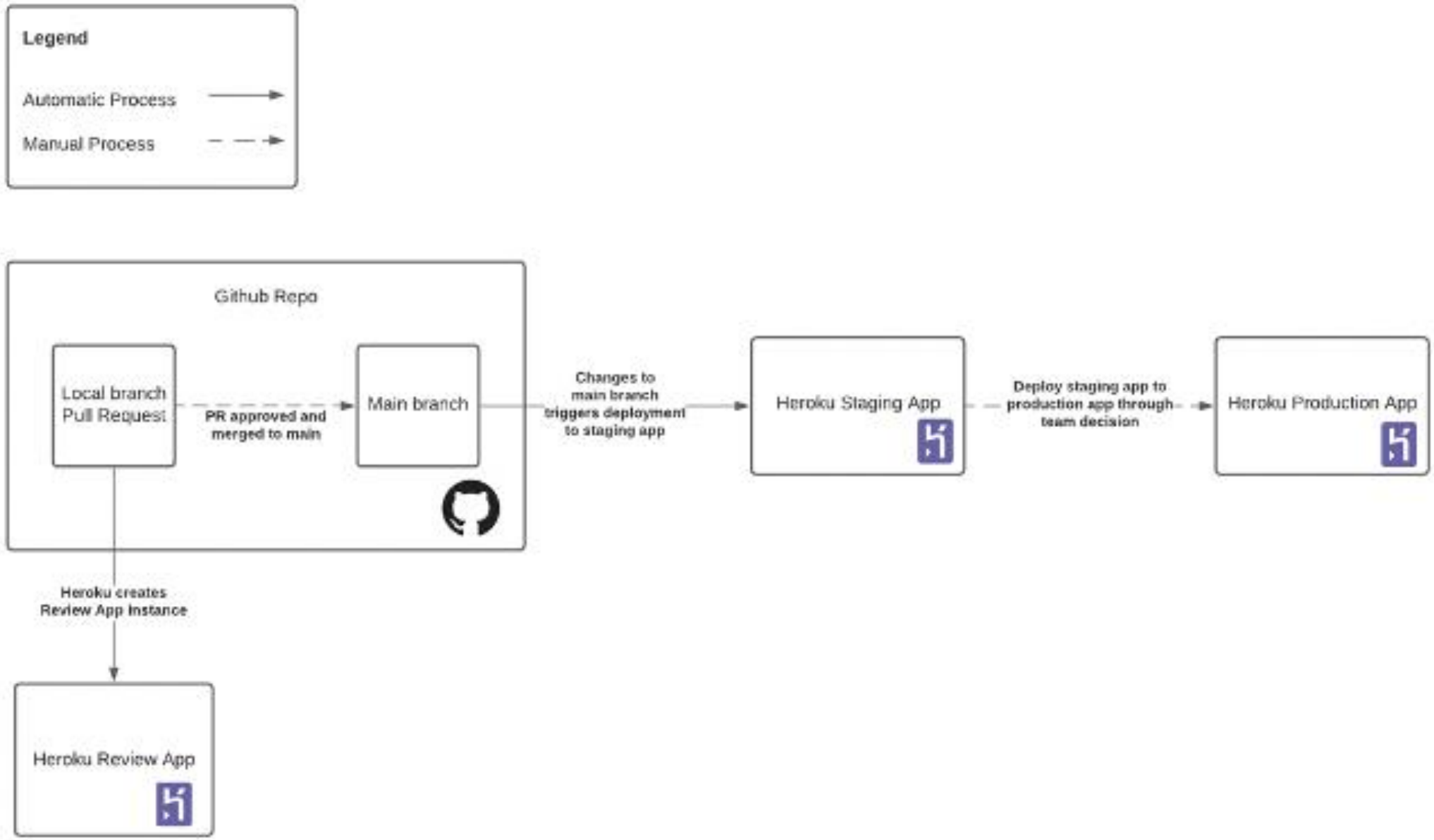


Physical View

- Depicts the system from an engineer's point of view
- Concerned with the topology of software components at the physical layer, as well as the physical connection between these components
- Represented using the deployment diagram

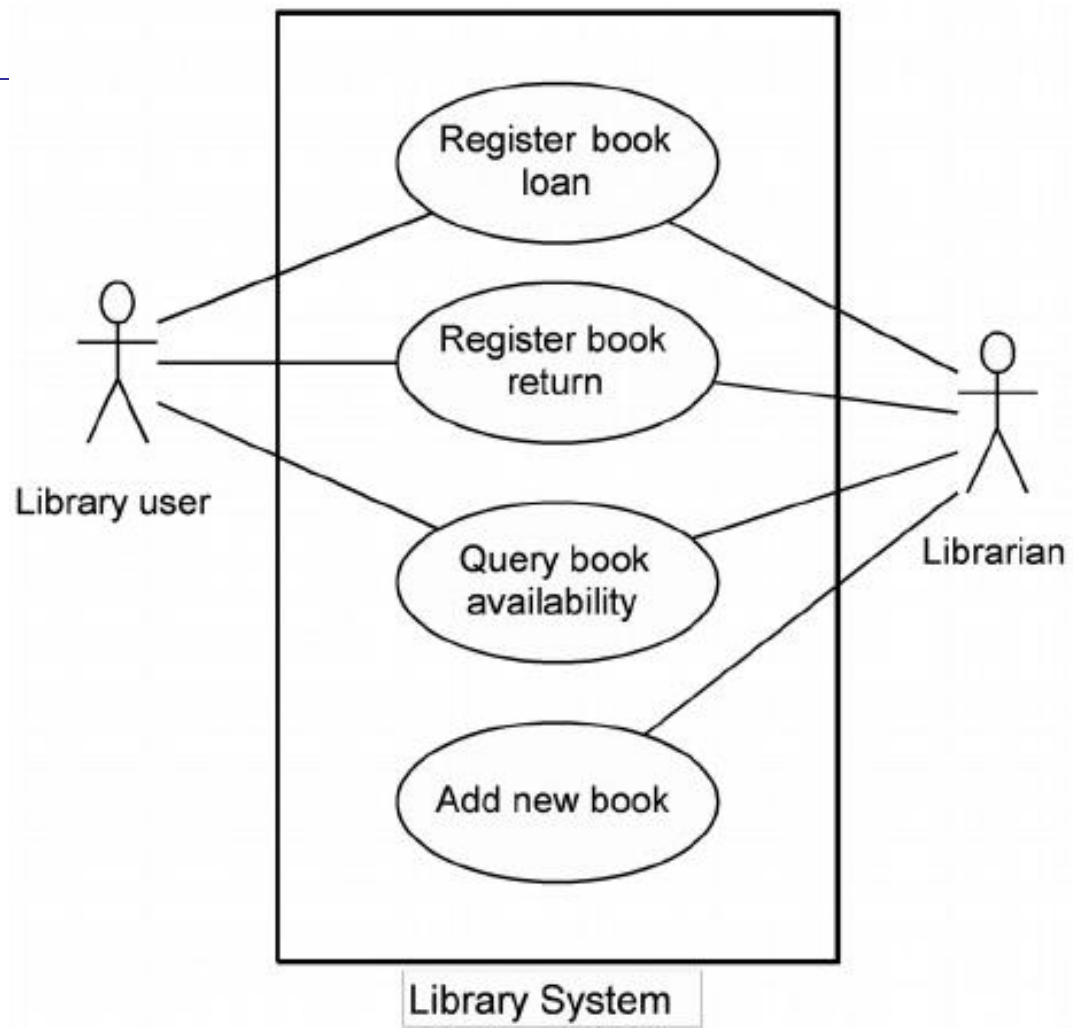


Deployment Pipeline | Lucidchart

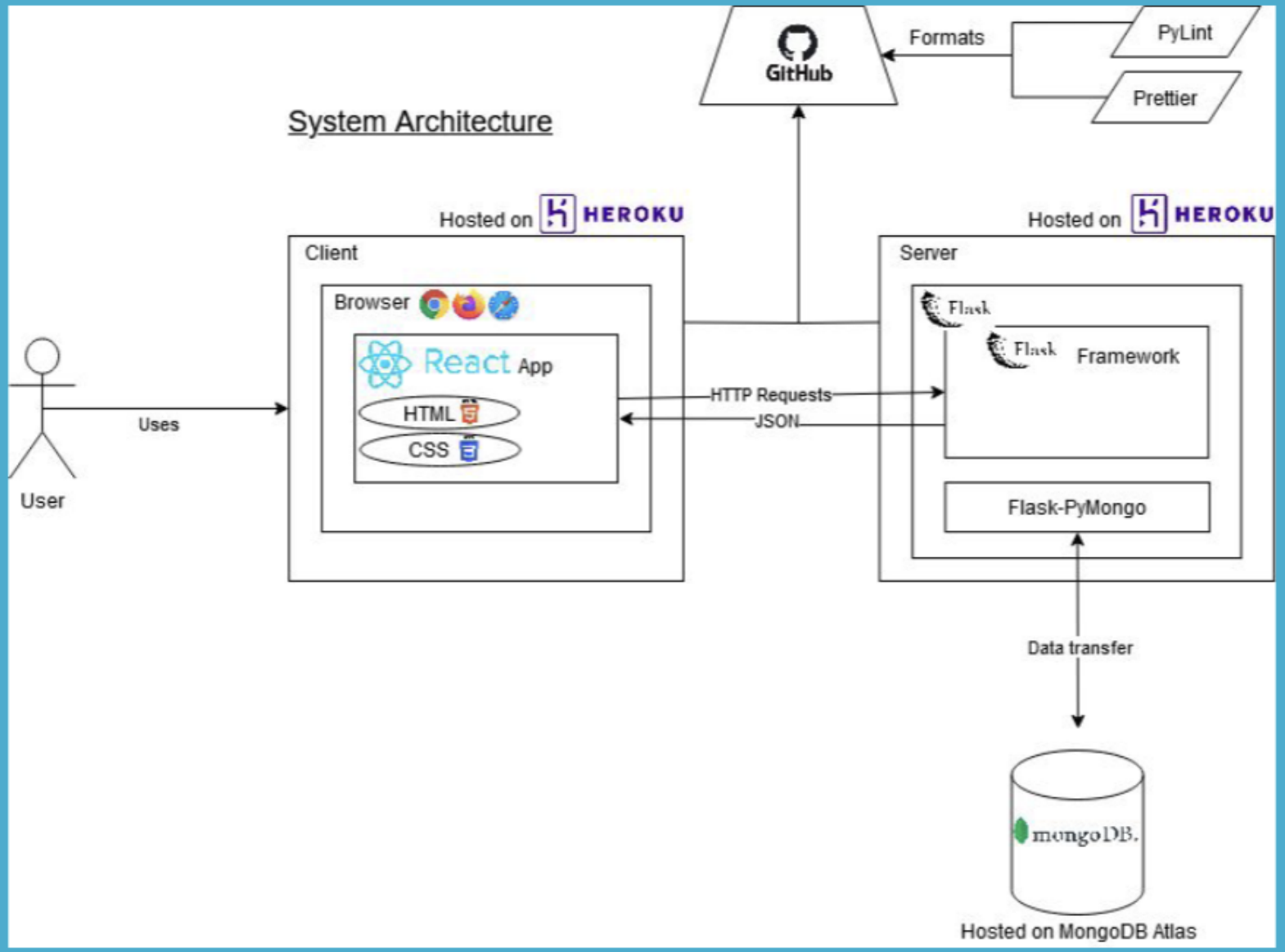


Scenario / Use Case View

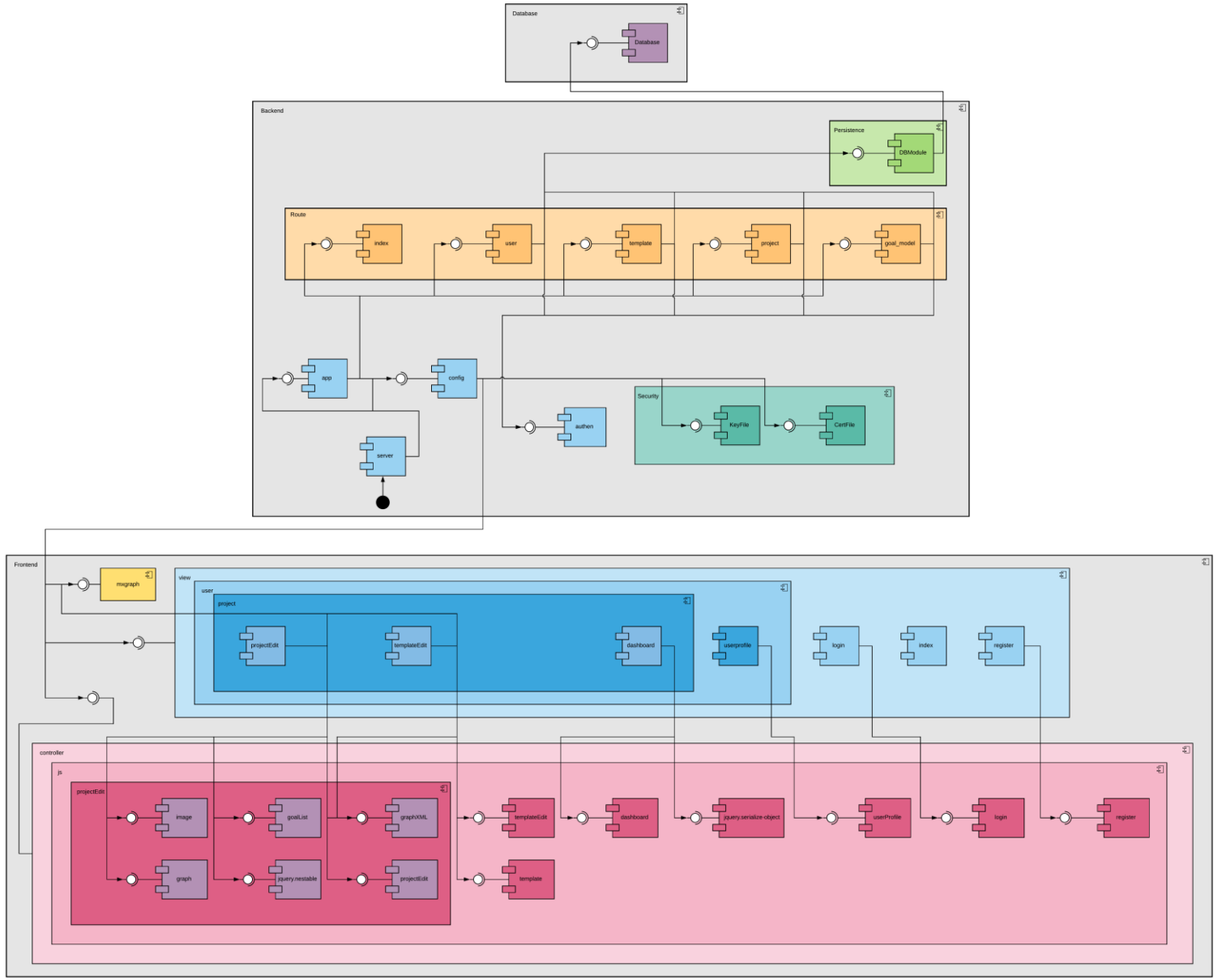
- Show a subset of important use cases
- Represented by use case diagrams



Architecture Diagram



MME design


Database
Database
Backend
Persistence
DBModule
Route
index
user
template
project
goal_model
app
config
authen
server
Security
KeyFile
CertFile
Frontend
mxgraph
view
user
project
projectEdit
templateEdit
dashboard
userprofile
login
index
register
controller
js
projectEdit
image
goal
goalList
jquery.nestable
graphXML
projectEdit
templateEdit
template
dashboard
jquery.serialize-object
userProfile
login
register

THE UNIVERSITY OF MELBOURNE

FRONTEND DESIGN

Low Fidelity Prototypes

- Quick and easy way to make sure interface is fit-for-purpose
- Only includes basic aspects of visual design (e.g. shapes of elements)
- Only includes key content elements
- No final colours, visual elements, etc…
- Test interactions with users

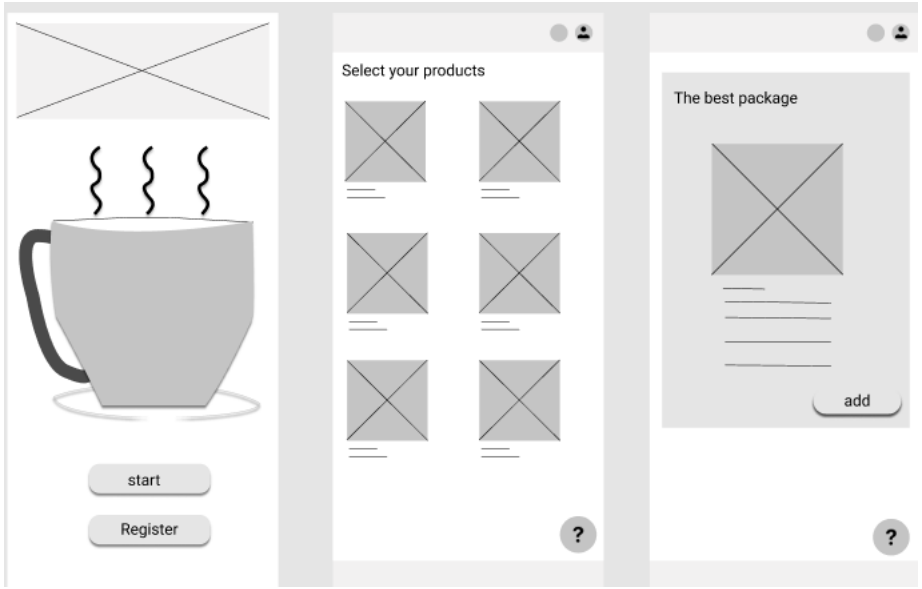


Image: Figma.com

High Fidelity Prototypes

- Appear as similar as possible to actual product
- Develop after you have a more solid understanding
- Include realistic content, visual design, colours, etc…
- Realistic interactions

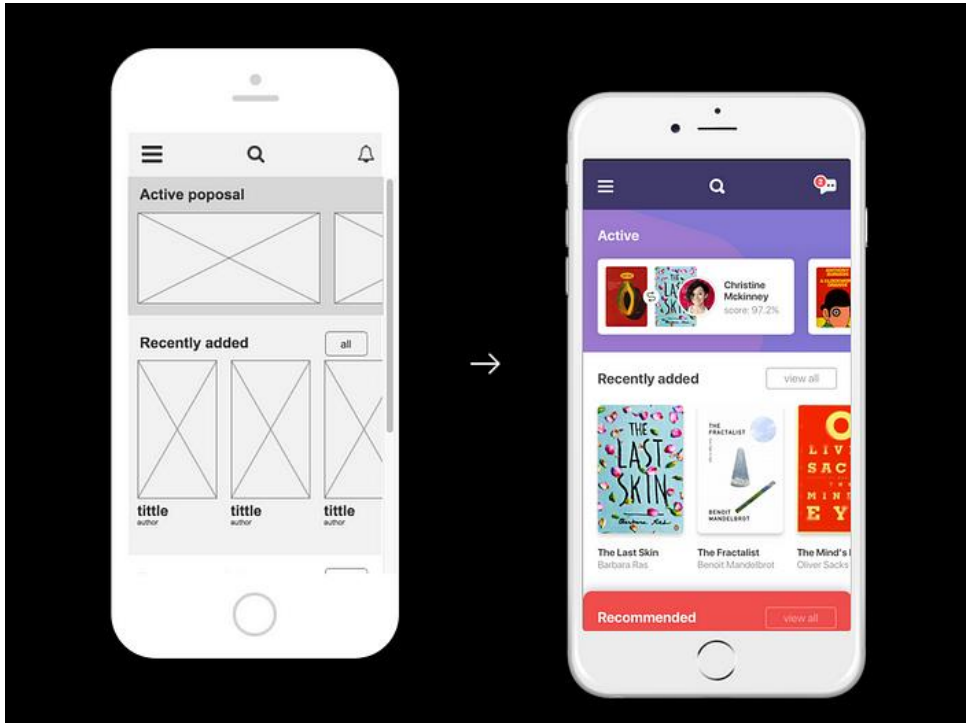


Image: https://medium.com/7ninjas/low-fidelity-vs-high-fidelity-prototypes-903a7befaa5a

Design in COMP30022

- Think about who you need to communicate with
- Think about what needs to be documented
- Review important documents with your team
- Use them to identify flaws in your design
- Keep them up to date
- Tools like Draw.IO or MS Visio are options for architecture
- Tools like Figma, Marvel, Canva are options for prototyping
- Discuss with your supervisor

Continuing design

- Past projects
  - We no longer write a 10 page project report
  - Some things are good in the project
- Great resource: https://cis-projects.github.io/project_based_course_notes/topics/devsprint.html