

Package

APPLICATION DEVELOPMENT FOR MOBILE AND UBIQUITOUS COMPUTING

TECHNISCHE UNIVERSITÄT DRESDEN

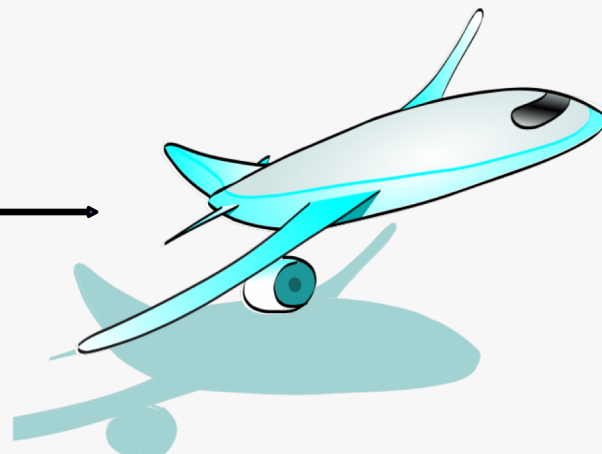
HUSSAM ALTALEB

&

AHMED MAHMOUD

SUPERVISED BY: DR. -ING. THOMAS SPRINGER

Trade Training



How Package solves the problem ?

- User can filter trips specifying group of people in lobby screen (realtime)
 - Max weight offered by the traveller to carry
 - Available Weight (weight available in the travel luggage).
 - Contact info such as name, phone and address for communication (can be copied from user profile).
- Trips are sorted according to the location of the user (nearest travellers show first).
 - Max weight offered by the traveller to carry
 - Available Weight (weight available in the travel luggage).
 - Contact info such as name, phone and address for communication (can be copied from user profile).



Challenges

- Power consumption.
- Data handling.
- User interface responsiveness.
- Network integration.

Adaptivity

- Location Based Search
- Offline usage in case of network outage
- Real time update

Location Based Search

- Learn about location from the context.
- Use that information to provide suggestions of journeys or queries where the current location is the origin of the journey.
- Keep location queries to a minimum to save power.

Offline usage in case of network outage

- Results of user queries are stored temporarily on the user device.
- Those results can be used as a cache:
 - Save Bandwidth
- They also can be used when user is offline:
 - Offline usage
 - Must denote somehow that user is offline, and when they are back online

Real time updates

- Scenario:
 - User **X** searches for journeys from **A** to **B** then user **Y** posts a journey from **A** to **B**.
- User **X** must be notified of the new posted journey.
- To be implemented using a Model-View-Controller (MVC) in the logic tier.

Architecture (Tiers) and Technologies

Data Tier Data Store

Django ORM to SQL

- Handling database queries in Python OO manner allowing for flexible logic development
- ~~MVC~~
- Community Driven

Logic Tier (Middleware) Interfacing (and decoupling) end tiers

Django REST framework

- Efficiently handle and serialize Django data
- ~~MVC~~
- Community Driven

Presentation Tier User Interface

React Native

- Cross Platform
- JavaScript
- Live reload
- Community Driven

Work Plan: Tasks

Hussam

- Data modeling
- Data interfacing
- MVC
- *Others as needed!*

Ahmed

- User interface
- Consuming REST API inside the application
- *Others as needed!*

Work Plan: Milestones *(tentative!)*

- By the second presentation deadline:
 - Implement MVC's and adaptation concepts and their underlying infrastructure
- By final presentation deadline:
 - Finalize the application and implement unit tests for it