PARKMAN

PARKING SPACE MANAGEMENT SYSTEM

Project in subject "Agile Software Development (MTAT.03.295)"

Team members:

Bejon Sarker (B79616)

Grace Achenyo (B79599)

Navedanjum Ansari (B46022)

Rando Tonisson

Course Instructor:

Luciano García-Bañuelos

Table of Contents

1. Description of Solution

In this section the feature solution, application architecture, snapshots of the solution and some of the challenges faced is presented briefly.

1.1 Features

In this section, we briefly describe the features implemented and the screenshot from the application in the snapshot view section of the report will gives a clearer idea of how the user interface looks like and functionality in general.

1. **User Sign up feature:** New users can register themselves to use the system by filling in the registration page that includes setting up a username and password for login into the system
2. **Login and Logout feature:** Users can login to the system with their login credentials and at anytime they can logout from the system. We have login and logout buttons to conduct these user actions from the UI
3. **Search parking space feature:** Users can search the parking spaces available based on the current location and he gives the estimate of time duration for which parking space is needed.
4. **Price schemes:** User can mouse hover on any of the available or displayed parking space to get the summary of price schemes such as hourly rate, real time rate, estimated hourly price and estimated real time price for that parking space choice.
5. **Book parking space feature:** parking space can be booked by clicking on the available choices on the city map.
6. **Choose payment options feature:** Once a parking space is booked users can choose to pay now or pay later and they also have a choice to select either of the payment schemes i.e. hourly or real time.
7. **Extend time notification feature:** Hourly scheme users receive a notification before the end time if they want to extend using the parking time by an hour. User can choose to extend the time or leave as it is to end using the parking space.
8. **Booking summary feature:** Users can view the list of booking by navigating to the Booking page.
9. **Pay from Booking page:** User can make the payment for the unpaid bills by clicking on the payment action option, displayed against the unpaid row to make the payment. Payment is made from the user's credit card information stored in the system in the encrypted form.
10. **Update Account information feature:** Parkman users can update their account information by navigating to the account page and update the information in the system. Once they edit the information and click on Update button, the updated information is stored in the systems database.

1.2 Architecture

We tried to use the hexagonal architecture. For the Front end we use Quasar framework, Backend we use phoenix framework and database is PostgreSQL. We have used Vue JS and Jquery in the implementation of front-end.

1.3 Snapshot views

In this section, you will find the snapshot view of the application features covering the parking scenario and implemented functionalities.

- User Registration or Sign up

Personal Information:

Naved

Name

naved

Username

•••••

Password

Payment Card Details:

navedanjum

Cardholder Name:

1234567890123456

Card Number:

•••

CVC:

1220

Expiry Date (mmyy) :

→] SIGNUP

Have an account? Log in

- Login to the system

naved

Username   5 / 10

•••••   👁

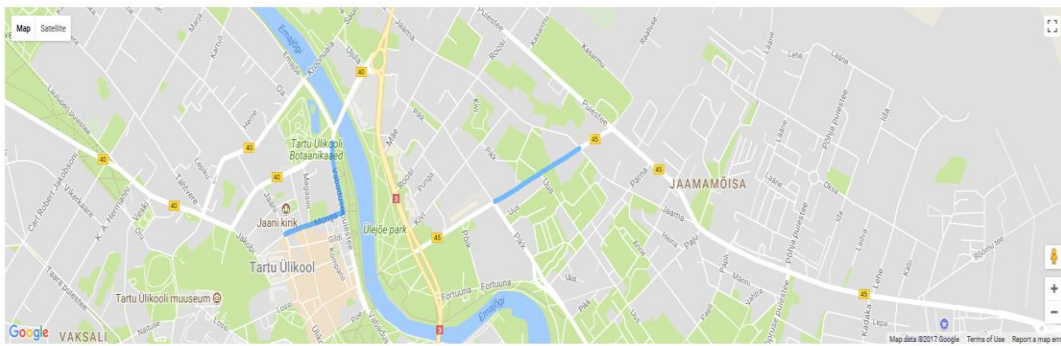Password   5 / 10

→] LOG IN

New to here? Sign up now

- Search for parking space

- Available parking spaces are displayed
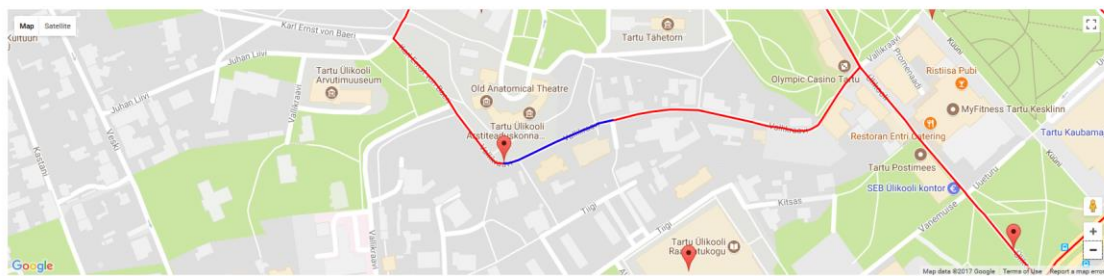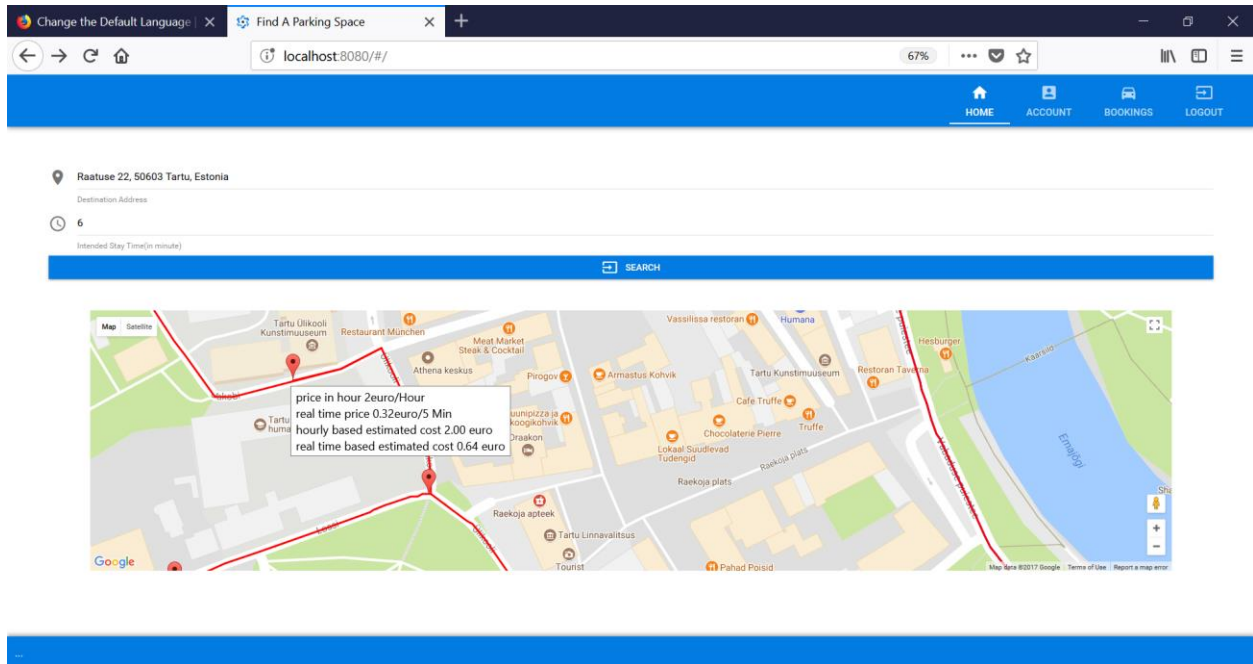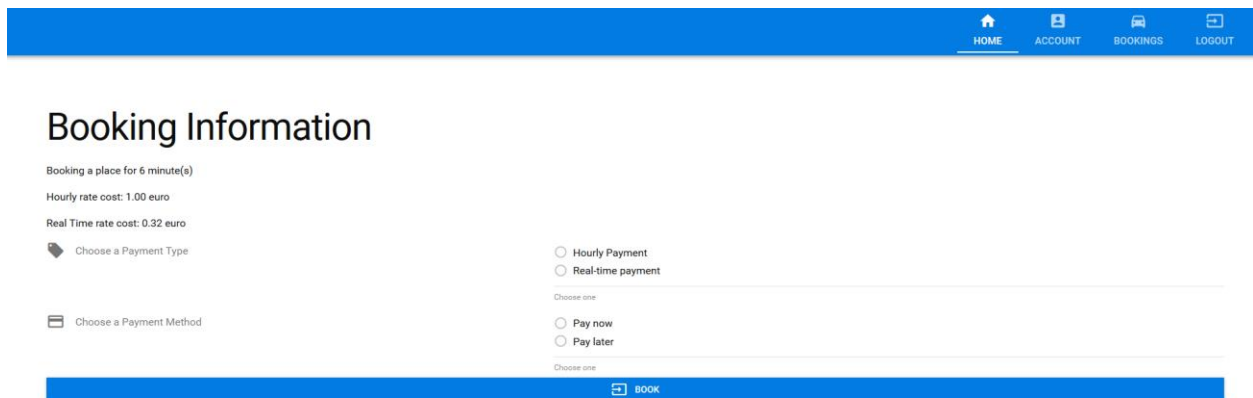
- Hovering mouse over the parking location on map will display the price schemes for the parking space

- Click on the paring location on map to book which will display the Booking information and options to choose the payment schemes and payment options

- Select the payment scheme and choose to Pay Now or Pay later

- Customers get Notification to extend the booked parking space.
- They can extend the parking time for an hour or reject it

- On top right, clicking on Account tab will let customers update their account details. Once updated click Update button to save the updated information in systems database

- Parking space booking history is displayed when Bookings Tab in the top right is clicked

- Parking space Booking tab has a feature to make payment for the parking bills unpaid
- Clicking on payment Action against unpaid bill will make the payment from the available credit card details for the user in the Parkman system

Parking Space Bookings

	Parking Date	Parking Time (minutes)	Cost of Parking	Payment Type	Payment Status	Payment Action
☐	12/19/2017, 11:41:17 AM	6	1	HOURLY	PAID	✏
☐	12/19/2017, 11:41:48 AM	10	0.32	REAL_TIME	PAID	✏
☐	12/19/2017, 11:44:36 AM	6	2		PAID	✏
☐	12/19/2017, 12:08:27 PM	6	2		UNPAID	✏

Rows 15 ▾   1 - 4 / 4 |< <   1 / 1   > >|

Payment Successful!   ✕

- Logout from the system
- At any time, users can logout from the system by clicking on Logout tab at the extreme top right on the user interface.

1.4 Challenges Faced

Some of the challenges we faced are:
- Initial set-up and configuration for the application
- Integration of Quasar front-end and Phoenix backend
- Debugging some the issues were time consuming and challenging

2. Agile Practices in Action

As a first step towards the development process, all the requirements have been gathered and understood to deliver a quality product. We have used Trello for tracking and using the agile methodology.

All the features are mapped to the user stories on the Trello dashboard.

- **Kick off Meeting:** Initially we have a meeting to come up with the list of all the possible tasks and user stories.
- **Product Backlog:** We created product backlog consisting of all the feature stories and the tasks.
- **Sprint:** We have a weekly sprint to deliver a shippable feature. On *trello dashboard,* we picked up features to be implemented in each of the sprints. In total, we implemented the project in four sprints.
- **Sprint Backlog:** At the end of every sprint end, we moved the selected feature stories to the next Sprint and continue to work on it.
- **Sprint meeting:** The weekly sprint meeting is conducted in the practice sessions on every Thursdays. We ensure that everyone attend the meeting and show the shippable working product to the supervisor.
- **Continuous Integration:** The addition of every feature in sprints ensure that the code is working, and previously shipped feature is working as expected by way of manual test execution.
- **Daily Scrum:** we have a daily discussion for 5 minutes addressing where each of the team members answer three questions
  1. Which task he or she is currently working on?
  2. Which task he is planning to do next?
  3. Any impediments or issue that is preventing the scrum team member to complete the task?

For discussion purpose, Facebook messenger group was created so that each of us stay informed project related activities like delay in implementation, technical or managerial help etc.

The overall use of agile practices in this project help us to deliver the product in short span of time and it was easy to track the progress. With each sprint, the progress is clearly visible in the form of a piece of working application.

3. Development Platforms

The code repository is supported in **bitbucket** source code management system. The agile practices have been carried out with the help of **Trello dashboard.** The entire code has been written with the help of Visual studio code in Elixir, using Phoenix framework, quasar framework and vue JS. The database used is Postgres.

Links:

Bitbucket: https://bitbucket.org/account/signin/?next=/graceokolo/find-a-parking-space/overview

Trello Dashboard: https://trello.com/b/affMp5yJ/development-board

Pair programming Video: https://youtu.be/2wFTL19J4CU

4. References and Others

1. Postgres database documentation: https://www.postgresql.org/docs/

2. Phoenix Framework: http://phoenixframework.org/

3. Quasar Framework: http://quasar-framework.org/

4. Elixir documentation: http://elixir-lang.github.io/docs.html

5. Jquery: https://jquery.com/

6. Vue JS: https://vuejs.org/v2/guide/