|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| PROGRESS REPORT 1  Uservices – Service Provider Finder  **Project Members**  Marcelo Meirelles Boldo – 300314616  Section: 090  [meirellesboldom@student.douglascollege.ca](mailto:meirellesboldom@student.douglascollege.ca)  Juliana Yumiko Takase – 300302956  Section: 091  takasej@student.douglascollege.ca |
| |  |  |  | | --- | --- | --- | | Vancouver, June 2021 |  | Douglas College 4270 – Applied Research | |

1. **Title**

Application Name: Uservices

1. **Executive Summary**

Every day we face unexpected setbacks. A broken hydraulic pipe, a washing machine that stopped working. People work all day in the office and do not have time to go for a walk with their pet dog. Sometimes we only need an extra hand to make a residence move.

In addition to having to deal with these setbacks, the process of looking for a service provider is not always an easy task to perform.

To help ordinary people deal with this type of problem, Uservices is the ideal tool. Uservices aims to bring all service providers together on one platform.

Uservices intend to gather people how need services with people that can provide services.

1. **Project Justification**
   1. **Problem Statement**

Every day ordinary people face setbacks that take time to manage, regardless of their complexity.

A person who needs a particular service usually looks in the neighbourhood newspaper or search engine to find the specific service provider to eliminate their problem.

When searching the internet, for example, several recommendations are made by the tool, but this suggestion is not always the most appropriate since the first sites that the tool returns are paid sites to appear as first options.

Additionally, logging on to various websites and trying to guess which service provider will do the best service at the best price is tiring work.

To help ordinary people to solve their misuse in a more agile and secure way, Uservices will bring all service providers together on one platform.

With Uservices, users will be able to search for service providers, compare prices, see service providers with the best rating and recommendations, schedule services for the most convenient day and time and pay for the service in a safe and guaranteed way.

* 1. **Target Groups**

Uservices will benefit people from all walks of life and economic backgrounds:

* Small, medium, and large business service providers
* Independent Service Providers
* Common worker
* Families in general
* Small businesses
  1. **Research**

To help in the development of Uservices, we will carry out surveys with different types of target groups.

In addition to the features, we intend to implement, we will ask the respondents for suggestions, improvements, or changes in the Application.

We will carry surveys into steps, before development and after development with a test application.

1. **Project Implementation**
   1. **Technologies to be used**

**Web front-end:** React.js, Bootstrap

**Backend:** Express.js

**Database:** Mongo DB

**Hosting:** AWS EC2

**Justification:** As discussed at the June 3rd meeting, we decided to implement a web platform using React.

* 1. **Use Case Diagram**

**Diagram

Description automatically generated**

* 1. **Activity Diagrams**

We developed the Activities Diagram for the following activities:

1. Login User and Service Provider: The login screen will be unique, we will handle the login of users without distinction of the type of user, being user or service provider. The user profile will be differentiated after login according to the type of user that has been registered.

Diagram

Description automatically generated

1. Register User: Ordinary users will have a simple registration screen, with their basic data such as name, email and address

Diagram

Description automatically generated

1. Register Service Provider: on the registration screen the user can choose to be a Service Provider and will have to provide documents for background checking.

Diagram

Description automatically generated

1. Create Service Provider Profile: The service provider will have a profile screen to fill out. This information will describe the services that will be provided, the price charged and the profile picture. Each service must enter a type of category pre-established by the system.

The Service Provider profile will be subject to approval by Uservices.

Diagram

Description automatically generated

1. Create Service Provider Work Schedule: The Service Provider will have to maintain a work schedule where he will establish which days and hours he will work. On this screen, the Service Provider can choose to accept recursive jobs, that is, a user can contract the service periodically.

Diagram

Description automatically generated

* 1. **Database Structure**

This is a preliminary structure of the database. May change as the project progresses.

* + 1. Users:
       - Name
       - Surname
       - Email
       - Address
       - Hired Service Provider
         1. Rating
         2. Comment
       - Service Provider (bool)
         1. Services (Object Array)
         2. Service Provider description
         3. Background Documents
         4. Calendar
         5. Work Schedule

Days

Time

Accepts recursive jobs

* + 1. Services:
       - Category Name
       - Service Title
       - Service Description
       - Pictures
       - Price
    2. Categories:
       - Category Name (Example: Pet, Health, Maintenance)
       - Category Description

1. **Progress Table**

|  |  |  |
| --- | --- | --- |
| **Activities** | **Due** | **Completion** |
| Project proposal elaboration and submission | May 10 to May 14 | **ü** |
| Preliminary project scope | May 10 to May 14 | **ü** |
| Use Case Diagram sketch | May 10 to May 14 | **ü** |
| JavaScript and React preliminary study | May 10 to May 14 | **ü** |
| Project scope decision | May 17 to May 28 | **ü** |
| Macro Use Case Diagram | May 17 to May 28 | **ü** |
| Preliminary Database structure | May 17 to May 28 | **ü** |
| React.js, React Native, Node.js labs and study | May 17 to May 28 | **ü** |
| UC1 and UC2 specification | May 31 to June 11 | **ü** |
| Activity diagrams for UC1 and UC2 | May 31 to June 11 | **ü** |
| Coding and testing of UC1, UC2 | May 31 to June 11 | **ü** |
| Report 1 creation and submission | June 4 | **ü** |
| UC3 and UC4 specification | June 14 to June 25 | **ü** |
| Activity diagrams for UC3 and UC4 | June 14 to June 25 | **ü** |
| Coding and testing of UC3, UC4 | June 14 to June 25 | **l** |
| Paper update to include work in progress | June 14 to June 25 | **l** |
| Video presentation creation and submission | June 14 to June 25 | **l** |
| UC5 and UC6 specification | June 28 to July 9 | **l** |
| Activity diagrams for UC5 and UC6 | June 28 to July 9 | **l** |
| Coding and testing of UC5, UC6 | June 28 to July 9 | **l** |
| Report 2 creation and submission | June 28 to July 9 | **l** |
| UC7 and UC8 specification | July 12 to July 23 | **l** |
| Activity diagrams for UC7 and UC8 | July 12 to July 23 | **l** |
| Coding and testing of UC7, UC8 | July 12 to July 23 | **l** |
| Report 3 creation and submission | July 12 to July 23 | **l** |
| UC9 and UC10 specification | July 26 to August 6 | **l** |
| Activity diagrams for UC9 and UC10 | July 26 to August 6 | **l** |
| Coding and testing of UC9, UC10 | July 26 to August 6 | **l** |
| Final report creation and submission | July 26 to August 6 | **l** |
|  |  |  |
| **Label** |  |  |
| Completed | **ü** |  |
| Overdue | û |  |
| Still Due | l |  |