**G**

**Project Plan Document**

**Version: 0.1**

Table of Contents

[1. Document History 4](#_Toc18134622)

[2. General Description 4](#_Toc18134623)

[3. Specification/Requirements 4](#_Toc18134624)

[3.1 Registration/Log In 4](#_Toc18134625)

[ Client 4](#_Toc18134626)

[3.2 Home Page 4](#_Toc18134627)

[ Client 4](#_Toc18134628)

[ Service 4](#_Toc18134629)

[3.3 Specialists’ list page 5](#_Toc18134630)

[ Client 5](#_Toc18134631)

[3.4 Specialist’s profile page 5](#_Toc18134632)

[ Client 5](#_Toc18134633)

[3.5 Service type 6](#_Toc18134634)

[ Client 6](#_Toc18134635)

[3.6 Book 6](#_Toc18134636)

[ Client 6](#_Toc18134637)

[3.7 Menu 6](#_Toc18134638)

[ Client 6](#_Toc18134639)

[ Service 6](#_Toc18134640)

[4. Architecture 7](#_Toc18134641)

[5. Open questions 7](#_Toc18134642)

[6. Tools used 8](#_Toc18134643)

[7. Useful info 8](#_Toc18134644)

[7.1 Links 8](#_Toc18134645)

1. Document History
2. General Description
3. Specification/Requirements

## 3.1 Registration/Log In

### Client

When a client runs the app he/she should be prompted either to register (Sign Up) or Log In. After selecting “Sign Up”, “Register” window will be prompted. It will contain the following required fields:

* Email
* Name
* Password
* Re-Enter Password
* Phone Number

In front of each field we have textbox in order to input appropriate data. After checking if all fields are filled correctly, client will press “Register” button. For safet, user will be prompted to enter verification code sent to the Email which was given earlier by the user. After checking the correctness of the verification code the user will be registered, otherwise the user will be notified that the code was wrong.

If user selected “Log in”, he/she should be able to fill “Username” and “Password” fields. Please, note that as a username we will consider Email entered in phase of registration. Also the possibility to change the password should exist. So, there will be “Forgot password” button (or link) which will led to receiving new Email and then the user can change password (maybe here we should send notification to the phone to make sure if the user is a right person).

## 3.2 Home Page

### Client

After logging in the user will see “Home Page” (or “Specialists”). It will represent all available specialists to the user. Specialists will be shown as an icon and appropriate text (e.g., doctor, hairdresser etc.) below that icon. If a specialization has a sub specialization (e.g., doctors can be dentists, neuropathologists etc.) the next window will show those sub specializations for the selected specialization.

### Service

Service provider here should see calendar similar to Google’s calendar, hence it should show time and a short info about a client who has been registered (reserved) in that time. Service app user should be able to see the calendar for today, week, month or a custom selected day. The clients’ info should be clickable and will show detailed information about them after clicking to the selected client. So, a new window will be shown. In the new opened window a service provider will see the following information about a client:

* Photo
* Data & Time reserved
* Name
* Phone
* Some notes about the client

Service provider should be able to edit some info about registered user, e.g., Date & Time reserved (fields like Photo, Name, Phone are readable) and send notification and/or Email to the user to approve changed Date & Time.

The provider also can add a client from scratch by clicking to the empty client info in the calendar.

## 3.3 Specialists’ list page

### Client

Here (name of the dialog is not determined) user will see all specific specialists representing in the window line by line. In each line will be short info about the service provider. The info should contain:

* Avatar
* Name
* Rating (represented by 5 stars)
* First available time
* Price (or price range)
* Location

It is possible that the price will be undefined.

Here user should be able to filter the specialists in a various of ways, e.g., by ratings, by price etc.

## 3.4 Specialist’s profile page

### Client

This dialog should represent more info about selected specialist. It should show avatar a little bit bigger, besides it should show:

* Name
* Rating
* Location
* Phone
* Short description about specialist (note: this info should be given by service provider itself in “EasyBusy Server” application)
* Cancellation Period

Cancellation period depends on the service provider and application somehow should be sure that the client saw this notification.

Then user can start booking process by clicking to “Book” button.

## 3.5 Service type

### Client

In many cases specialists provide different types of services. For example, hairdresser can only fan or cut hair in different ways. Or doctor can only consult or extract tooth etc. Hence, “Service Type” page will provide those info (either listing those line by line or by grid).

## 3.6 Book

### Client

In this dialog user should select Date and Time. Date will show enable dates of the concrete service provider. Other days should be disabled.

After selecting date user will select available time. Then user can either go back or book the service.

When booking process succeeds green mark will be shown on the screen. Also both user and service provider will be notified about the deal (it is preferable both by email and sending notification to the phone).

## 3.7 Menu

### Client

Menu (Settings) of the client will have the following fields:

* Favorites
* Calendar
* Profile
* Notifications
* Logout
* Calendar

Here user will see Google like calendar with booked/reserved days.

* Profile

User profile will contain *“avatar”, “phone”, “password”* and ability to change those.

### Service

Menu (Settings) of the service provider will have the following fields:

* Name
* My services
* Profile
* Notifications
* Logout

#### My services

In this window will be shown different types of services provided by the concrete service provider. By clicking ‘+’ button the provider can add another service in the newly opened window. Here the provider should specify:

* Specialization/Area:
* Actions: which types of sub services he/she provides (e.g., dentist can just consult or do other many things)
* Details: some detailed information about him/her
* Photo(s): certifications etc.

When selecting “Actions” the provider should provide name, duration (or range), price (or range) for the selected action. Also the provider will specify description of the action shortly and add some photo(s). Name of the action should be selected from the collection sent from DB.

#### Profile

Profile page represents detailed information about the service provider. It should contain avatar of the provider as well as the fields like:

* Name\*
* Specialization(s)\*
* Rating
* Location\*
* Work hours\*
* Details\*
* Documents/Photos

There should be possibility to edit fields by clicking ‘Edit’ button (or pencil).

#### Notifications

In this page a service provider will see different types of notifications, e.g., “book added” “book canceled”, “book changed”, “rating (review) received”. Also the provider will receive notification 10 minutes before meeting with a client.

1. Architecture
2. Open questions

* Should the user receive notification on phone for changing password?
* What should the name of the window listing all specific specialists?
* Should we show “Specialist’s Profile Page” every time? Or maybe it is possible to book immediately from “Specialists’ list Page”, because it is possible that the client is already familiar with the specialist and doesn’t want to know detailed information about him. In this case “Book” button should exist in “Specialists’ list page” as well.

1. Tools used
   * Linux – operating system
   * NetBeans – IDE for Java development
   * Android Studio – for Android emulator
   * VS Code – IDE for React Native development
   * Git - distributed version control system
   * GitHub – storage for the project
   * MongoDB - database
2. Useful info

## 7.1 Links

<https://egghead.io/> - Learning platform

<https://google.github.io/styleguide/cppguide.html> - coding style to be used