



Power Marketing

Robin Assistant
Technical Design - V1.0

Summary

1. Introduction:	3
1.1 Project overview	3
2. Coding and frameworks:.....	4
2.1. Client App:	4
2.2. Caretaker Dashboard:	4
3. Code Management	5
3.1 Git:	5
3.1 Git flow:	5
3.1 Trello:	5
4. Development Environment	6
4.1 Environment:	6
5. Activity diagram:	7
5.1 Dashboard:.....	7
6. Relationeel datamodel.....	8
6.1 App:.....	8

1. Introduction:

1.1 Project overview

Robin Assistant is a new assistive software solution for people with special needs and their careers in performing everyday activities. It is used for different types of brain conditions: autism spectrum, Down syndrome, Alzheimer's disease various forms of dementia, and other sorts of brain injuries.

This document will show all information about the web-app dashboard project plan for the caretakers and the activity-app for the clients.

2. Coding and frameworks:

2.1. Client App:

To create the app for the caretaker's client we are going to use Laravel. Laravel is a secure and useful framework for custom applications. It has many API functions which can be used for the notification's API.

When this Laravel app is complete it will be added into a Dart Flutter WebView element which will have a custom back end for receiving and sending notifications from the web API.

2.2. Caretaker Dashboard:

The caretaker's dashboard will also be created in Laravel. The default Authentication template from Laravel/UI bootstrap will be used. The bootstrap scaffolding will be overwritten by our custom SCSS and HTML structure.

3. Code Management

3.1 Git:

We are going to use Git for managing our code. The Git platform of choice will be GitHub. All code will be stored on GitHub and issues in the code can be reported on the issues page in GitHub, so the development team is notified about certain issues.

3.1 Git flow:

We will be using Git Flow, which is a custom module for Git that allows multiple developers to work on certain parts of the application at the same time without code loss or migration issues.

3.1 Trello:

When a new feature is published to Git there will be a Trello card moved to “For Review” where the other developers can see what is changed and allow the change to be published.

4. Development Environment

4.1 Environment:

Our development environment will contain the following software:

- Apache2 httpd
- PHP 8.0.2
- Composer 2.0.11
- NPM 7.6.0
- MySQL 8
- phpMyAdmin 5.0.4

The project will be connected to a Git repository which can be used to push changes from the developers and pull all the new features on the live server to update the application.

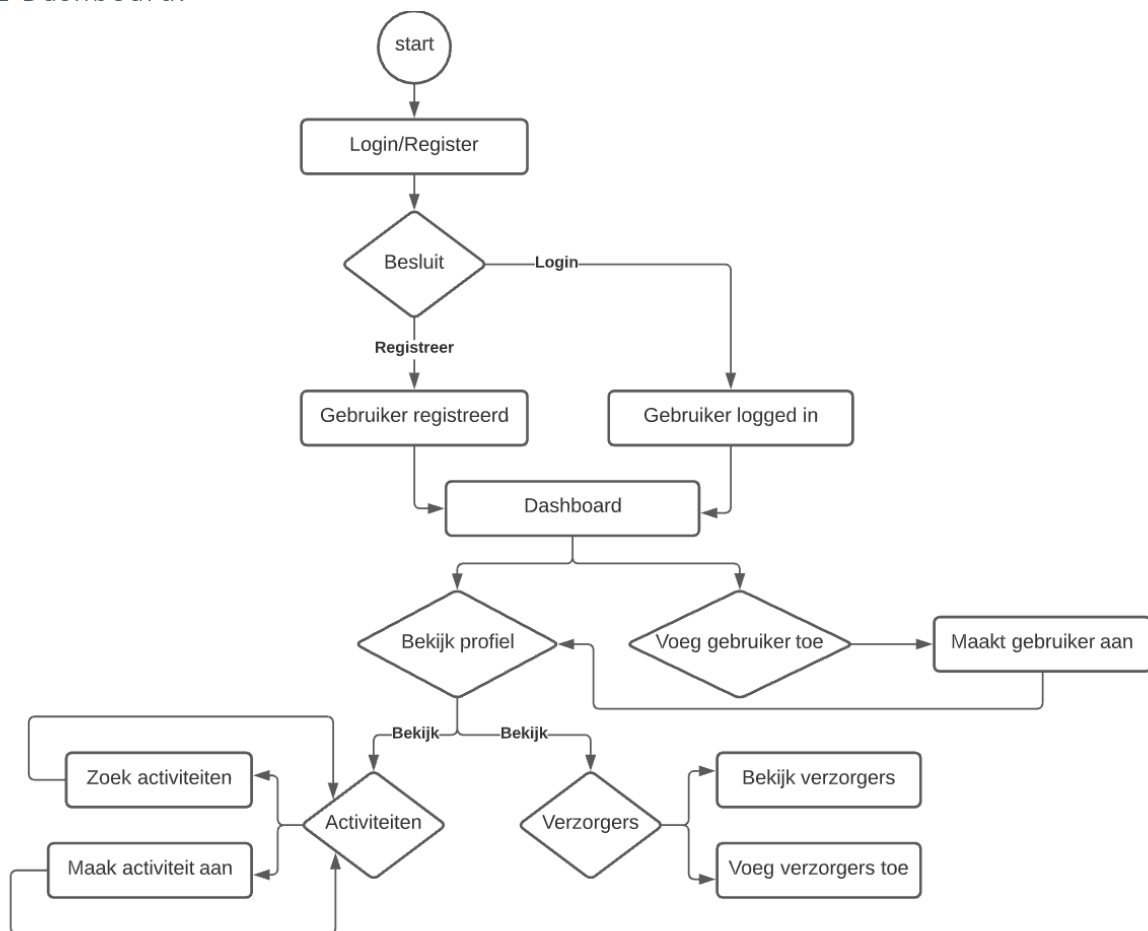
Every time you would like to update the Laravel application you would need to execute the following commands from SSH:

1. `cd project/folder`
2. `php artisan down`
3. `git pull`
4. `php artisan up`

- 1. Navigate to the folder the application is stored.*
- 2. Put the Laravel application in maintenance mode so users can't execute anything during the update.*
- 3. Retrieves all updates from the Git repository.*
- 4. Put the Laravel application back online for the users.*

5. Activity diagram:

5.1 Dashboard:



6. Relatieel datamodel

6.1 App:

6.1.1 Verzorgers datamodel:

Users:

ID	Name	Email	Password	Phone	Controlling
1	John Doe	j.doe@gmail.com	[REDACTED]	+31 6 38823914	2
2	Jane Doe	Jane.d@gmail.com	[REDACTED]	+31 6 49934025	1

(Unsigned bigint) **ID**: Verzorgers unieke ID

(varchar 255) **Name**: Verzorgers gebruikersnaam

(varchar 255) **Email**: Verzorgers email

(varchar 255) **Password**: Beveiligd password met PHP PASSWORD_DEFAULT algoritme

(varchar 255) **Phone**: Verzorgers telefoonnummer

(Unsigned bigint) **Controlling**: Welke gebruiker de verzorger bestuurd.

6.1.2 Gebruikers datamodel:

Clients:

ID	User_id	Firstname	Lastname	Birth	Profile_picture
1	2	Joe	Public	01-01-1976	https://x.com/
2	1	Tom	Bloggs	01-01-1985	https://x.com/

(Unsigned bigint) **ID**: Gebruikers unieke ID

(Unsigned bigint) **User_id**: Gebruikers zijn verzorgers unieke ID

(varchar 255) **Firstname**: Gebruikers voornaam

(varchar 255) **Lastname**: Gebruikers achternaam

(varchar 255) **Birth**: Gebruikers geboortedatum

(varchar 255) **Profile_picture**: Externe of interne link naar profiel foto.

Client_activities:

ID	Client_id	User_id	Name	Description	Begin_time	End_time	Icon_src	Video_src	Sound_src
1	2	1	Bath	Take a bath	15:01	15:15	https://x.com/	https://x.com/	https://x.com/
2	2	1	Toilet	Go to toilet	15:01	15:15	https://x.com/	https://x.com/	https://x.com/
3	1	2	Bath	Take a bath	15:01	15:15	https://x.com/	https://x.com/	https://x.com/

(Unsigned bigint) **ID**: Uniek taak ID

(Unsigned bigint) **Client_id**: Taak gebruikers unieke ID

(Unsigned bigint) **User_id**: Taak verzorgers unieke ID

(varchar 255) **Name**: Taak naam

(varchar 255) **Description**: Korte taak beschrijving

(datetime) **Begin_time**: Begin tijd van taak

(datetime) **End_time**: Eind tijd van taak

(varchar 255) **Icon_src**: Icoon source intern of extern locatie

(varchar 255) **Video_src**: Video source intern of extern locatie

(varchar 255) **Sound_src**: Sound source intern of extern locatie