Assignment 1	Project Summary
Course	Practical Robotics and Smart Things - 2020

Project author		
№	Pseudonym	Face-to-face/ online
1	maya	face-to-face

Project name	Pet Feeder

1. Short project description (Business needs and system features)

Nowadays IoT is facing great development. Its various applications range in all kind of human activities. The following project concentrates in its use in the everyday life, and more precisely pet care. The idea is to build smart widget that will take care of pet feeding throughout the day while the owner is not home.

The hardware implementation is based on Raspberry Pi 3 (Model B) single board computer, equipped with following sensors:

- *Raspberry Pi camera (v2)* - used to take user's pictures and send them to the face recognition service;
- *PIR motion detector* - activating the camera and the detection;

ICM actuators include:

- Continuous rotation servo – rotating the portion wheel

The ICM has the following main modes of operation:

- *Sleeping mode* – during which the pet feeder is inactive;
- *Dog recognition mode* – activated around feeding times and on owner demand;
- *Feeding mode* - activated when dog is recognized;

There will also be a mobile application through which the owner can monitor whether the pet has eaten its portion and to manually give additional one.

The main user roles (actors in UML) are:

- *Unregistered User* – his only option is to register;
- *Registered User* – can register a pet feeding machine, obtain information about current feeding process and manually feed pet from one of the registered machines;

2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
2.1. Register	Unregistered user can register in the system through the mobile application.	*Unregistered user*
2.2. Add feeding machine	*Registered user* can add a feeding machine to configure and monitor.	*Registered user*
2.3. Delete feeding machine	*Registered user* can remove feeding machines from the ones it has registered.	*Registered user*
2.4. Login	*Registered user* must log in the system so to identify himself.	*Registered user*
2.5. Manually feed	*Registered user* can choose to manually feed his pet from some of his registered feeding machines.	*Registered user*
2.6. Set feeding times	*Registered user* can set how many times and at what hour his pet must be fed.	*Registered user*

3. Main Views (Mobile Web App)

View name	Brief Descriptions	URI
1. Home	The entry page of the mobile app. It has a welcoming message and buttons navigating to the only possible pages a user can enter at that stage: the register page and the login page.	*/*
2. Feeding Machines	Contains listed created feeding machines for the current user. The user can enter each feeding machine by clicking on it.	*/machines*
3. Create a feeding machine	In this view the registered user can create new feeding machines and connect them with the corresponding device. Also, he can set feeding hours and dose.	*/create*
4. Feeding Machine Status	This view presents a singular feeding machine: its status, whether the pet has been fed, an option for manual feed, and an option to delete the concrete machine.	*/machine/{id}*
5. Register	In this view an Unregistered user can register himself by entering sufficient information.	*/register*
6. Login	This view present the user with a login form which is used to identify him/her in the system and enable him to use it.	*/login*

7. Profile	In this view a registered user can review and update his personal information.	/profile

4. API Resources (REST/SSE/WebSocket Backend)		
View name	**Brief Descriptions**	**URI**
4.1. **Users**	GET list of all users of the system	/api/users
4.2. **User**	GET, PUT, DELETE User Data for User with specified userId	/api/users/{userId}
4.3. **Login**	POST User Credentials (e-mail address and password) and perform user login.	/api/login
4.4. **Logout**	POST a logout request	/api/logout
4.5. **Register**	POST User data (email address, name, password) and create user account in the system corresponding to that data.	/api/register
4.6. **Machines**	GET all feeding machines for the user with id *userid*. POST Machine data to register a new feeding machine to this user.	/api/machines/{userid}
4.7. **Machine**	GET, PUT, DELETE machine data to operate over machine specified by *mid*.	/api/machine/{mid}

Resources:

[1] https://www.youtube.com/watch?v=sCOkuyH7CPo&t=259s – from here I got the basic idea for the portion wheel

[2] https://create.arduino.cc/projecthub/circuito-io-team/iot-pet-feeder-10a4f3?ref=tag&ref_id=pet%20feeder&offset=0 – information about what kind of servo is sufficient for my purposes and the motion detecting sensor