Bilkent University

Senior Design Project

Drector

High-Level Design Report

Shahriyar Mammadli, Nihad Azimli, Burak Özmen, Veysel Alperen Ceylan

Supervisor: Varol Akman

Jury Members: Çiğdem Gündüz Demir, İbrahim Körpeoğlu

Innovative Expert: Mustafa Sakalsız

Project Website: https://drectorapp.github.io/Drector/

Table of Contents

1. Introduction 5

1.1. Purpose of the system 5

1.2. Design goals 6

1.2.1. Availability 6

1.2.2. Efficiency 7

1.2.3. Platform compatibility 7

1.2.4. Privacy 7

1.2.5. Scalability 7

1.2.6. Usability 7

1.3. Definitions, acronyms and abbreviation 7

1.4. Overview 7

2. Proposed software architecture 8

2.1. Overview 8

2.2. Subsystem decomposition 8

2.2.1. Presentation Layer 9

2.2.2. Application Logic Layer 10

2.2.3. Data Storage Layer 10

2.3. Hardware/software mapping 10

2.4. Persistent Data Management 11

2.5. Access Control and Security 11

2.6. Global Software Control 11

2.7. Boundary Conditions 12

2.7.1. Initialization 12

2.7.2. Termination 12

2.7.3. Failure 13

3. Subsystem Services 13

3.1. User Interface Subsystem 13

3.2. User Management Subsystem 13

3.3. Image Processing Subsystem 14
3.4. Report Subsystem 14
3.5. Map Mode Subsystem 14
3.6. Data Management Subsystem 14
4. Glossary 15
5. References 16

Table of Figures

Figure 1 Subsystem Decomposition Scheme 9
Figure 2 Hardware and Software mapping 10

1. Introduction

Lately drones and quad copters started to get huge part of life in general. Almost in every industry people use drones or quad copters for different purposes and usage of these flying objects is increasing day by day in vast level. However, there is no mechanism to get information about the objects flying around us. As stated above, drones can be used for multiple purposes; these purposes consist of both good and bad ones. Therefore, there is a demand for registering drones or keeping information about them. It is fact that governments want to learn specifications of unknown drones. Even more, anyone has right to learn about undefined flying object around him/her. Unfortunately, there is no way to get information about them. We plan to develop mobile application, Drector that will solve this worldwide issue. This cross-platform application will show in-demand information about drones and quad copters.

Drector is a mobile application which will be implemented for both IOS and Android to keep track of drones and quad copters that surrounds you. Users will be able to reach the information about drones or quad copters using Drector which have two different modes. Users can inspect the drones or quad copters that are nearby with map mode or camera mode of Drector.

1.1. Purpose of the system

Drector is planned to be global drone tracking service which makes available of getting real time information about drones all around the world. There are 3 different user types in the system which additional features are provided due to their role. Normal user, owner, and authorities are the user types. System has functionalities for the normal user that provides tracing drones all around the world with the help of map. User can click on the specific drone and see the information about both drone and owner. Furthermore, the drones, is in specific distance from the current location of user, can be selected and traced by the help of the map. That is, system provides users with the functionality that distance manually can be selected to trace the drones. It is much more useful in the manner of seeing that if there is alien or unknown drone around the user. For example, user can easily trace and get information about the strange drone in his or her garden. Furthermore, there is a camera mode which user can track the drone by the help of camera of the mobile phone. It is main usage is for the possibilities of having more than one drone that surrounds user and he or she wants to get and information about the specific drone in certain direction. When drone is recognized in the camera user can click and get information of the drone and the owner. To enhance the project and eliminate the small possibilities such as having more than one drone in specific direction, meaning that multiple overlapping drones or multiple back to back drones, user can click specific drone in the camera and can obtain data about that particular drone. Camera functionality is also a solution for the situation that user has more than one drone around which causes confusion when trying to recognize specific one in the map. Most important feature of the system is to provide users with functionality of the reporting the unknown and suspicious drone to the authorities. It is also purposive tool for the authorities to handle the security issue that are caused by the drones. Second user type is the owner which is provided with the same functionalities as normal user, however, additionally, has get license and added own drone to the system. Additional, UI is provided for the owner which by the help of this interface user it can be seen that if the drone is reported or not. However, data about the reported will not be given. Furthermore, when user flying the drone and area that drone is flied is problematic because of frequent reports, then, user immediately gets notification about it. Lastly, authorities can use the functionalities as above-mentioned roles, however, other than that; they can check reports to initiate the investigation about it. Authorities are able to reach the information about both reporter and reported person and their drones. By the help of system, security matter about the drones will be improved and issues that caused by drones will be minimized.

1.2. Design goals

1.2.1. Availability

The application need to be ready to use in any time, thus, users could utilize the functionalities of the system.

1.2.2. Efficiency

The system needs to be efficient and thus, users could use functionalities fast. Especially, while catching the drone with the help of camera, system need to serve quickly not to miss data of the drone.

1.2.3. Platform compatibility

Project need to be compatible with the iOS and Android, therefore, the application will be developed in Angular 4.

1.2.4. Privacy

Drone owner's only name and surname will be seen by the other users even in the situation of reporting. Contact info and personal information will not be shared with other users. Only authorities can reach the contact information of the owners. Moreover, owners cannot reach the information of the users who reported them.

1.2.5. Scalability

To improve scalability unnecessary data will not be held in database, network chatters will be minimized, and caching will be used in possible situations.

1.2.6. Usability

Application will have best possible UI to provide users with practicality and simplicity.

1.3. Definitions, acronyms and abbreviation

Angular: is a Typescript-based open-source front-end web application platform led by the Angular Team at Google and by a community of individuals and corporations.

UI: user interface

1.4. Overview

Drector app is designed to minimize drones' both intentionally and unintentionally infringe of people's privacy and moreover, can be used in military level. It is intended to improve security issues that caused by drones. Application uses image processing to recognize drones by the help of camera. Users are provided with the feature of seeing previously reported drones. A user sees the drones which pinned in the map, however, due to frequent change of locations; map view is updated in every 5 seconds. User gets license for his or her own drone and by the help of this license, drone is registered in the system. This system is planned to fill the gap in the security which occurred due to drones and continuous technology development.

2. Proposed software architecture

2.1. Overview

In this part, we will decompose our system into maintainable subsystems. Our main aim is not only reducing the coupling between the different subsystems of the system, but also increasing the cohesion of subsystem components. In addition, we tried to decompose our system in order to apply client-server architectural style on our system.

2.2. Subsystem decomposition

In this part the subsystem decomposition of Drector is explained with a Figure 1.

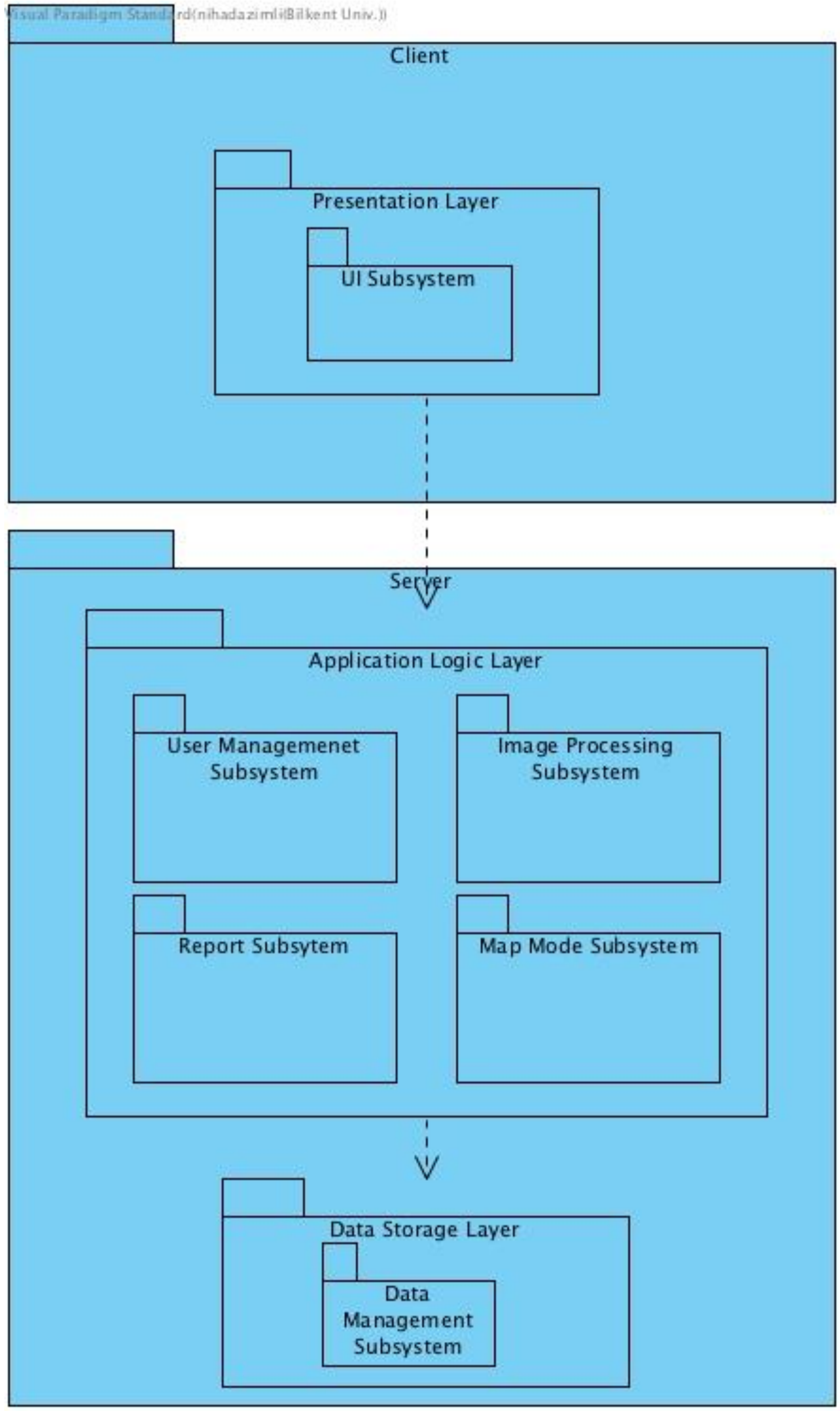
Visual Paradigm Standard(nihadazimli(Bilkent Univ.))
Client
Presentation Layer
UI Subsystem
Server
Application Logic Layer
User Managemenet Subsystem
Image Processing Subsystem
Report Subsytem
Map Mode Subsystem
Data Storage Layer
Data Management Subsystem

Figure 1 Subsystem Decomposition Scheme

2.2.1. Presentation Layer

This layer is where user encounters and interacts with. Presentation layer is only part of client subsystem. The layer interacts with Application layer through requests of

Presentation Layer. This layer interacts with Server subsystem, that is why, sustainability and security must be ensured.

2.2.2. Application Logic Layer

Application layer is core of our system. This layer is where our business logic. There is four subsystems under application layer. These are User Management Subsystem, Image Processing Subsystem, Report Subsystem, and Map Mode Subsystem. Subsystems mentioned above must act according to users' interactions.

2.2.3. Data Storage Layer

This layer will be composed of Data Management Subsystem where all of user data, reports and other necessary data. Application Layer will receive information from Application Logic Layer and ask or add demanded information to Data Storage layer

2.3. Hardware/software mapping

Drector is an application that will be developed for both Android and IOS operating system so that most of the smartphone users can use it. That is why user will have Android or IOS mobile phone. On client side of the system client will run JavaScript on the background. If user executes some operation related to Application Server it will go to web server through HTTP request and if needed it sends HTTP request to Database which runs MySQL in background.

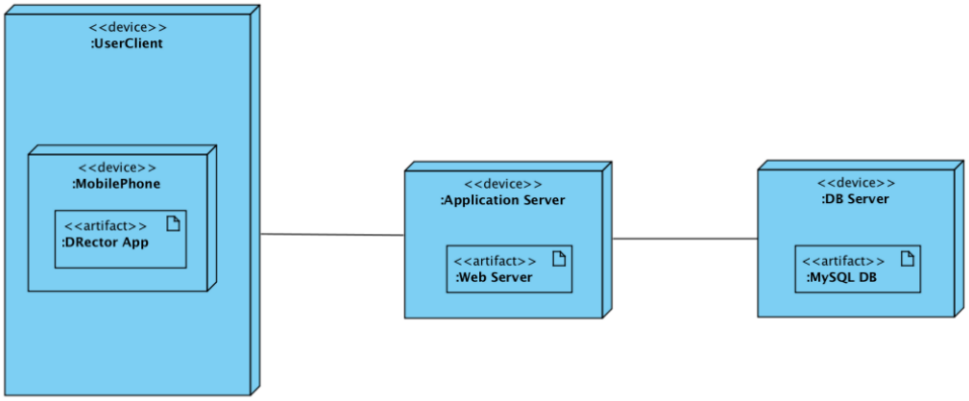

Figure 2 Hardware and Software mapping

2.4. Persistent Data Management

In order to store and manipulate users' and drones' information in an organized way, a NoSQL database will be used. Besides common information of users and drones, current location of drones will be stored in our database. Therefore, performance of database system is crucial for our system. A NoSQL database will be a good choice for our system because of its good performance with massive volumes of new, rapidly changing data. Also, a NoSQL database works well for applications that contain services which must be always-on, accessible from many different devices and scaled globally to many users. These are the reasons why NoSQL database will be used to store information consistently.

2.5. Access Control and Security

Users should provide a full name, a username, an e-mail address and a password when they are trying to register to the system. The e-mail addresses should be valid and unique, and the username should be unique. Since the user information will be kept in the database, the database will be used to check the uniqueness of the e-mail address and username in the process of registration. System will allow or deny the user according to the information he provides when he tries to register. Therefore, user accounts are protected by users' login information. Also, Drector will not allow to login after three wrong attempts of login process. On the other hand, Drector users will have an option to hide some of their information. The hidden information of the user will not be seen by other users.

2.6. Global Software Control

Drector is a client-server application and in Drector system, the global software control behaviors are event-driven. The current location of all drones are kept in the database and updated frequently. Along with location of drones, other information of drones are kept in the database and refreshed according to changes. Data flow between client and server side is triggered by adding a new drone to the system, registering to the system, any change of the information of users or drones, or detecting a drone using camera or map mode of Drector. In the case of detecting a drone, data flow starts from client side to server side. The data, which is collected while detecting a drone, will be sent to server, and after processing the data, the server will send a response to the client. The information of the detected drone will be showed to the user.

2.7. Boundary Conditions

Boundary conditions have three parts:

- Initialization
- Termination
- Failure

2.7.1. Initialization

In this part, the initialization of the application will be evaluated.

- Users must download the application from Google Play Store or App Store.
- Users must start the application by clicking its icon on their phones.
- Users must have an internet connection on their phones in order to use the application.
- Users must register to the system before using the application.
- Users must login to the system after registration.

After completing all these steps, Drector's initialization part is completed and users can Drector easily.

2.7.2. Termination

In this part, the termination of the application will be evaluated.

The user can terminate the application using either home button or back button of their phones.

- In order to terminate the program using back button, the user should be on the map page, otherwise the user will be directed to the page before that. If the user is on the map page and press back button, a pop-up window will appear and asks user that if he desires to quit the application. In order to terminate the program, the user must select "YES".
- If the user closes the application using home button and does not clear his recently used application, the application will continue to work where he left when he opens the application again.

- If remember me option is selected while logging to the system, the user will not be required to login after he terminates the application. If remember me option is not selected while logging to the system, the user will be required to login after he terminates the application.
- In the case of updates, all the users that are logged in must login to the system after the update has been downloaded.

2.7.3. Failure

In this part, the reaction of system will be evaluated in the case of failure.

- If a problem occurs while downloading or updating the application, Drector will not be opened since its files have been corrupted.
- If there is an internet connection problem while using Drector, the application cannot perform the desired tasks since the system will not give any response.

3. Subsystem Services

3.1. User Interface Subsystem

This subsystem provides interface services to the user by providing a connection between other subsystems in Application Logic Layer. It displays the data which is available to the user from the other subsystems such as login page, sign up page, forgot password page, map view page, camera view page, information page, report page, profile page, and add drone page. The reason why we divided this subsystem as a separate part of the system is dealing with the complexity and lowering the coupling between subsystems.

3.2. User Management Subsystem

In this subsystem, the user information is processed and managed. Its login, signup, information about user, and adding drones of the system is done at this subsystem. Also, user commands are processed and transferred to the other subsystems at the same layer and other layers.

3.3. Image Processing Subsystem

Camera view is processed and managed, and the detection of drone with image processing is done at this subsystem. Also, data are transferred from the data management subsystem and transferred to the other subsystems at the same later and other layers.

3.4. Report Subsystem

In this subsystem, the reporting drones is processed and managed. The reporting to the officials and administrators the selected drone's information is done at this subsystem. Also, data are transferred from the data management subsystem and transferred to the other subsystems at the same later and other layers.

3.5. Map Mode Subsystem

In this subsystem, placing drones' location in map consistently is processed and managed and selection the drones from map is done. Also, data are transferred from the data management subsystem and transferred to the other subsystems at the same later and other layers.

3.6. Data Management Subsystem

Data Management Subsystem is located at the lowest layer of the system which is Data Storage Layer. This subsystem stores the data collected from upper layer and the outer system. In database, the users' and drones' information are stored.

4. Glossary

NoSQL: non-SQL(Structured Query Language)

Android OS: Android Operating System

GPS: Global Pointing System

IOS: Iphone Operating System

GSM: Global System for Mobile Communications

Angular: is a Typescript-based open-source front-end web application platform led by the Angular Team at Google and by a community of individuals and corporations.

UI: User interface

5. References

[1]Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.

[2]"NoSQL Databases Explained", MongoDB, 2017. [Online]. Available: https://www.mongodb.com/nosql-explained. [Accessed: 22- Dec- 2017].

[3]"Serving Web Content with Spring MVC", Spring.io, 2017. [Online]. Available: https://spring.io/guides/gs/serving-web-content/. [Accessed: 22- Dec- 2017].

[4]N. Agnihotri, "GSM/GPRS Module", Engineersgarage.com, 2017. [Online]. Available: https://www.engineersgarage.com/articles/gsm-gprs-modules. [Accessed: 22- Dec- 2017].

[5]"Angular Docs", Angular.io, 2017. [Online]. Available: https://angular.io/guide/quickstart. [Accessed: 22- Dec- 2017].

[6]"Ionic Framework", Ionic Framework, 2017. [Online]. Available: https://ionicframework.com/docs/. [Accessed: 22- Dec- 2017].

[7]Heeseo Chae, JunHo Park, HoNam Song, YongHwan Kim and HaeWook Jeong, "The IoT based automate landing system of a drone for the round-the-clock surveillance solution", 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2015.

[8]J. Kim, Y. Lee, S. Han, S. Kim, G. Lee, H. Ji, H. Choi and K. Choi, "Autonomous flight system using marker recognition on drone", 2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV), 2015.

[9]P. Daponte, L. De Vito, G. Mazzilli, F. Picariello, S. Rapuano and M. Riccio, "Metrology for drone and drone for metrology: Measurement systems on small civilian drones", 2015 IEEE Metrology for Aerospace (MetroAeroSpace), 2015.