

**iCatch  
Project handover plan**

**Version <1.0>**

## Release history

<b>Date</b>	<b>Version</b>	<b>Comment</b>	<b>Author (Developer)</b>	<b>Approved By (TL)</b>
N.A.	N.A.	N.A.	N.A.	N.A.

## Index

1.	Project background	4
	1.1 Client	4
	1.2 Purpose	4
	1.3 Scope	4
	1.4 Work role	4
2.	Project description	4
	2.1 Core work flow	4
	2.2 Architecture	4
	2.3 Work item	4
	2.4 Project Team	5
3.	Handover plan	5
	3.1 Handover timeline	5
	3.2 Handover method	5
4.	Notice	5
	4.1 Cooperation model	5
	4.2 Frequently Asked Questions	5
5.	Account information	5
	5.1 Site	5
	5.2 Source code	5
	5.3 Resources	5
	5.4 App iTunes/Play Store URL	5
6.	Work Status	5
	6.1 Current Status	5
	6.2 Status project on last day of contract	5

# Project handover plan

## 1. Project background

iCatch is a social app based on Fishing in which user can create and join a fishing tournament and submit the catches he got in that particular tournament.

To create tournaments user has to specify the start date-time, end date-time of the tournament, the location where the tournament will take place and the species for which the participants can compete.

In order to submit images of catch, he just have to click the image from the custom camera used in the app, adjust the draggable pointers for fish head, tail and scalar pointer and submit. Length and weight of the fish is calculated according to the pointers.

Basically iCatch is an online app, but in case user doesnot have internet connection, he can still submit the catches which will be saved in the database and will be added to the tournament once the user is online again and opens the app.

User can login into the app using facebook and share there posts on popular social networks like facebook, instagram and twitter. He/she can also invite people using email or through facebook.

Retrofit 2.0 has been used to implement the web-services.

### 1.1 Client

N.A.

### 1.2 Purpose

*To manage the fishing tournaments*

### 1.3 Scope

### 1.4 Work role

*Development*

## 2. Project description

### 2.1 Core work flow

*Waterfall model was followed.*

### 2.2 Architecture

*MVC*

### 2.2 Work item

*Project is completed and delivered.*

## **2.3 Project Team**

P. Ravikant, Tarak Bhawalkar, Sneha Maheshwari, Ajinkya Saswade

## **3. Handover plan**

### **3.1 Handover timeline**

*N.A.*

### **3.2 Handover method**

*N.A.*

## **4. Notice**

### **4.1 Cooperation model**

*N.A.*

### **4.2 Frequently Asked Questions**

*N.A.*

## **5. Account information**

### **5.1 Site**

<https://www.icatchapp.com/>

### **5.2 Source code**

<http://tarak.bhawalkar@180.149.243.244:8080/scm/git/iCatch>

### **5.3 Resources**

*N.A.*

### **5.4 App iTunes/Play Store URL**

*N.A.*

## **6. Work Status**

### **6.1 Current Status**

*Completed*

### **6.2 Status project on last day of contract**

*Completed and Delivered*

**UBI**  
**Project handover plan**

**Version <1.0>**

## Release history

<b>Date</b>	<b>Version</b>	<b>Comment</b>	<b>Author (Developer)</b>	<b>Approved By (TL)</b>
N.A.	N.A.	N.A.	N.A.	N.A.

## Index

1.	Project background	9
	1.1 Client	9
	1.2 Purpose	9
	1.3 Scope	9
	1.4 Work role	9
2.	Project description	9
	2.1 Core work flow	9
	2.2 Architecture	9
	2.3 Work item	9
	2.4 Project Team	9
3.	Handover plan	9
	3.1 Handover timeline	9
	3.2 Handover method	9
4.	Notice	10
	4.1 Cooperation model	10
	4.2 Frequently Asked Questions	10
5.	Account information	10
	5.1 Site	10
	5.2 Source code	10
	5.3 Resources	10
	5.4 App iTunes/Play Store URL	10
6.	Work Status	10
	6.1 Current Status	10
	6.2 Status project on last day of contract	10



# Project handover plan

## 1. Project background

UBI is food ordering system which has two different app for clients and restaurant owners namely UBI and UBI Commerce.

UBI is the app which people will use to order food, from the restaurant listed in app, select delivery address and pay for food once it's delivered. UBI Commerce is for restaurant owners from which they can add restaurants, food categories and items they provide and its description.

### 1.1 Client

N.A.

### 1.2 Purpose

*To place and manage orders in a restaurant.*

### 1.3 Scope

### 1.4 Work role

*Development*

## 2. Project description

### 2.1 Core work flow

*Waterfall model was followed.*

### 2.2 Architecture

*MVC*

### 2.3 Work item

*Project is completed and delivered.*

### 2.4 Project Team

Tarak Bhawalkar

## 3 Handover plan

### 3.1 Handover timeline

N.A.

### 3.2 Handover method

N.A.

## **4 Notice**

### **4.1 Cooperation model**

*N.A.*

### **4.2 Frequently Asked Questions**

*N.A.*

## **5 Account information**

### **5.1 Site**

<https://play.google.com/store/apps/details?id=com.ubiapp>

### **5.2 Source code**

<http://mobilegit.neosofttech.in/Android/ubi.git>

### **5.3 Resources**

*N.A.*

### **5.4 App iTunes/Play Store URL**

*N.A.*

## **6 Work Status**

### **6.1 Current Status**

*Completed*

### **6.2 Status project on last day of contract**

*Completed and Delivered*

**IGIS**  
**Project handover plan**

**Version <1.0>**

## Release history

<b>Date</b>	<b>Version</b>	<b>Comment</b>	<b>Author (Developer)</b>	<b>Approved By (TL)</b>
N.A.	N.A.	N.A.	N.A.	N.A.

## Index

1.	Project background	15
	1.1 Client	15
	1.2 Purpose	15
	1.3 Scope	15
	1.4 Work role	15
2.	Project description	15
	2.1 Core work flow	15
	2.2 Architecture	15
	2.3 Work item	15
	2.4 Project Team	15
3.	Handover plan	15
	3.1 Handover timeline	15
	3.2 Handover method	15
4.	Notice	16
	4.1 Cooperation model	16
	4.2 Frequently Asked Questions	16
5.	Account information	16
	5.1 Site	16
	5.2 Source code	16
	5.3 Resources	16
	5.4 App iTunes/Play Store URL	16
6.	Work Status	16
	6.1 Current Status	16
	6.2 Status project on last day of contract	16

## Project handover plan

### 2. Project background

Mobile application that will help user get information related to their home including all the assets in it. This application will act as medium between rent payers, house renting company and technicians. Application will assist rent payers, to get automatic notifications related to assets as he/she comes near any beacon (attached to each asset) in home. User can get offer notifications and news feeds related to home which is posted by the house renting company. User can report issues related to assets in home (if any problem occurs) which will be resolved by technicians assigned to them. User can call emergency numbers related to each asset in case of any emergency. Mobile application is intended to be multilingual (German and English).

#### 1.1 Client

Alfons Parovszky

#### 1.2 Purpose

*To manage real estate properties.*

#### 1.3 Scope

#### 1.4 Work role

*Development*

### 2. Project description

#### 2.1 Core work flow

*Waterfall model was followed.*

#### 6.2 Architecture

*MVC*

#### 2.3 Work item

*Project is completed and delivered.*

#### 2.4 Project Team

Tarak Bhawalkar

### 3 Handover plan

#### 3.1 Handover timeline

*N.A.*

#### 3.2 Handover method

*N.A.*

## **7 Notice**

### **4.1 Cooperation model**

*N.A.*

### **4.2 Frequently Asked Questions**

*N.A.*

## **8 Account information**

### **5.1 Site**

*N.A.*

### **8.2 Source code**

[http://mobilegit.neosofttech.in/Android/igis\\_android.git](http://mobilegit.neosofttech.in/Android/igis_android.git)

### **8.3 Resources**

*N.A.*

### **8.4 App iTunes/Play Store URL**

*N.A.*

## **9 Work Status**

### **6.1 Current Status**

*Completed*

### **6.2 Status project on last day of contract**

*Completed and Delivered*

**World Of Fashion  
Project handover plan**

**Version <1.0>**



## Release history

<b>Date</b>	<b>Version</b>	<b>Comment</b>	<b>Author (Developer)</b>	<b>Approved By (TL)</b>
N.A.	N.A.	N.A.	N.A.	N.A.

## Index

1.	Project background	20
	1.1 Client	20
	1.2 Purpose	20
	1.3 Scope	20
	1.4 Work role	20
2.	Project description	20
	2.1 Core work flow	20
	2.2 Architecture	20
	2.3 Work item	20
	2.4 Project Team	20
3.	Handover plan	20
	3.1 Handover timeline	20
	3.2 Handover method	20
4.	Notice	21
	4.1 Cooperation model	21
	4.2 Frequently Asked Questions	21
5.	Account information	21
	5.1 Site	21
	5.2 Source code	21
	5.3 Resources	21
	5.4 App iTunes/Play Store URL	21
6.	Work Status	21
	6.1 Current Status	21
	6.2 Status project on last day of contract	21

## Project handover plan

### 3. Project background

World of Fashion is an independent module of MOE specially designed for the Event "Simply Dubai" held at the Mall of the Emirates. It has features to book tickets of the events and to see the media gallery of the events.

#### 1.1 Client

Mall of Emirates

#### 1.2 Purpose

*To show the event details in the mall.*

#### 1.3 Scope

#### 1.4 Work role

*Development, Client coordination*

## 2. Project description

### 2.1 Core work flow

*Waterfall model was followed.*

### 9.2 Architecture

*MVC*

### 2.3 Work item

*Project is completed and delivered.*

### 2.4 Project Team

Tarak Bhawalkar

## 3 Handover plan

### 3.1 Handover timeline

*N.A.*

### 3.2 Handover method

*N.A.*

## **4. Notice**

### **4.1 Cooperation model**

*N.A.*

### **4.2 Frequently Asked Questions**

*N.A.*

## **5. Account information**

### **5.1 Site**

<https://www.malloftheemirates.com/>

### **5.2 Source code**

<http://mobilegit.neosofttech.in/Android/old-moe.git>

### **5.3 Resources**

*N.A.*

### **5.4 App iTunes/Play Store URL**

<https://play.google.com/store/apps/details?id=com.belongi.moe>

## **6. Work Status**

### **6.1 Current Status**

*Completed*

### **6.2 Status project on last day of contract**

*Completed and Delivered*

**Win Win  
Project handover plan**

**Version <1.0>**

## Release history

Date	Version	Comment	Author (Developer)	Approved By (TL)
N.A.	N.A.	N.A.	N.A.	N.A.

## Index

1.	Project background	25
	1.1 Client	25
	1.2 Purpose	25
	1.3 Scope	25
	1.4 Work role	25
2.	Project description	25
	2.1 Core work flow	25
	2.2 Architecture	25
	2.3 Work item	25
	2.4 Project Team	25
3.	Handover plan	25
	3.1 Handover timeline	25
	3.2 Handover method	25
4.	Notice	21
	4.1 Cooperation model	21
	4.2 Frequently Asked Questions	21
5.	Account information	21
	5.1 Site	21
	5.2 Source code	21
	5.3 Resources	21
	5.4 App iTunes/Play Store URL	21
6.	Work Status	21
	6.1 Current Status	21
	6.2 Status project on last day of contract	21

## Project handover plan

### 4. Project background

The win-win application will allow a business user to upload video advertisements by recording advertisements in the application only and can post this advertisement after making required. The Business user can schedule advertisements for a particular time period. Uploaded advertisements will appear in the customer application. Customers will be able to view this advertisement by clicking on it and can share advertisements with their contacts.

#### 1.1 Client

Mall of Emirates

#### 1.2 Purpose

*To show the event details in the mall.*

#### 1.3 Scope

#### 1.4 Work role

*Development, Maintenance, Client coordination*

### 2. Project description

#### 2.1 Core work flow

*Waterfall model was followed.*

#### 9.3 Architecture

*MVP*

#### 2.3 Work item

*Project is completed and delivered.*

#### 2.4 Project Team

Tarak Bhawalkar

### 3 Handover plan

#### 3.1 Handover timeline

*N.A.*

#### 3.2 Handover method

*N.A.*



## **4. Notice**

### **4.1 Cooperation model**

*N.A.*

### **4.2 Frequently Asked Questions**

*N.A.*

## **5. Account information**

### **5.1 Site**

*N.A.*

### **5.5 Source code**

<http://mobilegit.neosofttech.in/Android/old-moe.git>

### **5.6 Resources**

*N.A.*

### **5.7 App iTunes/Play Store URL**

*N.A.*

## **6. Work Status**

### **6.1 Current Status**

*Completed*

### **6.2 Status project on last day of contract**

*Completed and Delivered*

**Plexigo  
Project handover plan**

**Version <1.0>**

## Release history

<b>Date</b>	<b>Version</b>	<b>Date</b>	<b>Author (Developer)</b>	<b>Approved By (TL)</b>
Plexigo Mobile	3.9.1	08/02/2022	Tarak Bhawalkar	Nitin Nohani
Plexigo Android TV	2.4	31/12/2021	Tarak Bhawalkar	Nitin Nohani
Plexigo FireTV	2.4	29/04/2022	Tarak Bhawalkar	Nitin Nohani
Plexigo Jio TV	3.3	20/01/2022	Tarak Bhawalkar	Nitin Nohani
Bharat OTT Mobile	1.12	30/05/2022	Tarak Bhawalkar	Nitin Nohani
Bharat OTT TV	1.2	06/04/2022	Tarak Bhawalkar	Nitin Nohani
TFS Play Mobile	1.19	25/05/2022	Tarak Bhawalkar	Nitin Nohani
TFS TV Launcher	14.0	31/05/2022	Tarak Bhawalkar	Nitin Nohani

## Index

1.	Project background	30
	1.1 Client	30
	1.2 Purpose	30
	1.3 Scope	30
	1.4 Work role	30
2.	Project description	30
	2.1 Core work flow	30
	2.2 Architecture	30
	2.3 Work item	30
	2.4 Project Team	30
3.	Handover plan	30
	3.1 Handover timeline	30
	3.2 Handover method	30
4.	Notice	31
	4.1 Cooperation model	31
	4.2 Frequently Asked Questions	31
5.	Account information	31
	5.1 Site	31
	5.2 Source code	31
	5.3 Resources	31
	5.4 App iTunes/Play Store URL	31
6.	Work Status	31
	6.1 Current Status	31
	6.2 Status project on last day of contract	31

# Project handover plan

## 5. Project background

Plexigo is a Video On Demand platform for Bollywood and Indian Regional movies. It provides a catalog of movies, series, and events available on different OTT platforms. Plexigo's own HD streaming service offers premium movie content if it is not available in theatres near you.

### 1.1 Client

Valuable Group

### 1.2 Purpose

*To Play OTT content owned by UFO as PVOD content and list the content available on other OTT Platforms.*

### 1.3 Scope

*OTT App*

### 1.4 Work role

*Development, Maintenance, Client coordination*

## 2. Project description

### 2.1 Core work flow

*Waterfall model is followed.*

### 9.4 Architecture

*MVC*

### 2.3 Work item

*Project is stable, enhancement work going on.*

### 2.4 Project Team

Tarak Bhawalkar (Android)

Abhijeet Gawde (iOS)

Komal Kolhe (QA Manager)

Medha Patankar (Android)

Ganesh Phatangare

## 3 Handover plan

### 3.1 Handover timeline

*2 Weeks*

### 3.2 Handover method

*Screen wise walkthrough of all apps with business logic*

## 4. Notice

### 4.1 Cooperation model

### 4.2 Frequently Asked Questions

Total Apps: 3

Plexigo: Android mobile, Android TV, Jio TV, Fire TV

Bharat OTT: Android Mobile, Android TV

TFS: Android Mobile, TV Launcher App

## 5. Account information

### 5.1 Site

<https://plexigo.com/>

<https://bharatott.co.in/>

### 5.8 Source code

<https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/plexigo>

[https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/PlexigoTV\\_android\\_app](https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/PlexigoTV_android_app)

[https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Bharat\\_OTT](https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Bharat_OTT)

[https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Bharat\\_OTT\\_TV](https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Bharat_OTT_TV)

[https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/tfs\\_plexigo](https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/tfs_plexigo)

[https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/TFS\\_PLEXIGO\\_TV](https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/TFS_PLEXIGO_TV)

### 5.9 Resources

Design Links :

<https://xd.adobe.com/view/0ceacbb7-f080-46d9-858c-a86d4c90c717-59b9/>

<https://xd.adobe.com/view/3ef66ea9-221f-43c1-a8b6-506a74ef0d84-6ba5/>

Shared Folder

[https://valuablegroup-my.sharepoint.com/personal/diit\\_dusaneinfotech\\_com/\\_layouts/15/onedrive.aspx?ga=1](https://valuablegroup-my.sharepoint.com/personal/diit_dusaneinfotech_com/_layouts/15/onedrive.aspx?ga=1)

### 5.10 App iTunes/Play Store URL

<https://play.google.com/store/apps/details?id=com.release.plexigo>

<https://play.google.com/store/apps/details?id=com.ott.bharat>

<https://play.google.com/store/apps/details?id=in.fiberstory.tfsott>

## 6. Work Status

### 6.1 Current Status

Underdevelopment, maintenance and enhancements tasks

**6.2 Status project on last day of contract**

All stable code shared with new developer, and given the access to all the documents required.

**DishButler  
Project handover plan**

**Version <1.0>**



## Release history

Date	Version	Comment	Author (Developer)	Approved By (TL)
N.A.	N.A.	N.A.	N.A.	N.A.

## Index

1.	Project background	35
	1.1 Client	35
	1.2 Purpose	35
	1.3 Scope	35
	1.4 Work role	35
2.	Project description	35
	2.1 Core work flow	35
	2.2 Architecture	35
	2.3 Work item	35
	2.4 Project Team	35
3.	Handover plan	35
	3.1 Handover timeline	35
	3.2 Handover method	35
4.	Notice	36
	4.1 Cooperation model	36
	4.2 Frequently Asked Questions	36
5.	Account information	36
	5.1 Site	36
	5.2 Source code	36
	5.3 Resources	36
	5.4 App iTunes/Play Store URL	36
6.	Work Status	36
	6.1 Current Status	36
	6.2 Status project on last day of contract	36

# Project handover plan

## 1. Project background

Dishbutler Admin app is a POS printer based application designed for restaurant admin. The SDK for the printer is provided by the manufacturer itself.

Whenever an order is placed at a particular restaurant, the admin is notified using FCM. The restaurant admin will approve the order and send it to the kitchen for preparation. For orders that need to be delivered or picked up ASAP, the admin can also select the time in which the order will be ready.

Once the order accepted, the customer gets notified about the order confirmation by SMS and also the time at which it will be ready.

Admin can also generate reports based on the time frame, users, and the platform from which the order is placed(web/app).

### 1.1 Client

Philippe Morgenthaler

### 1.2 Purpose

*To manage the orders placed at any restaurant*

### 1.3 Scope

*Hotel Management*

### 1.4 Work role

*Development, Maintenance, Client coordination*

## 2. Project description

### 2.1 Core work flow

*Waterfall model was followed.*

### 9.5 Architecture

*MVC*

### 2.3 Work item

*Project is completed and delivered.*

### 2.4 Project Team

Tarak Bhawalkar

## 3 Handover plan

### 3.1 Handover timeline

*N.A.*

### 3.2 Handover method

*N.A.*

## **4. Notice**

### **4.1 Cooperation model**

*N.A.*

### **4.2 Frequently Asked Questions**

*N.A.*

## **5. Account information**

### **5.1 Site**

*N.A.*

#### **5.11 Source code**

<http://mobilegit.neosofttech.in/Android/dishbutleradmin.git>

#### **5.12 Resources**

*N.A.*

#### **5.13 App iTunes/Play Store URL**

*N.A.*

## **6. Work Status**

### **6.1 Current Status**

*Completed*

### **6.2 Status project on last day of contract**

*Completed and Delivered*

# **Digital Inspection Project handover plan**

**Version <1.0>**

## Release history

Date	Version	Comment	Author (Developer)	Approved By (TL)
N.A.	N.A.	N.A.	N.A.	N.A.

## Index

1.	Project background	40
	1.1 Client	40
	1.2 Purpose	40
	1.3 Scope	40
	1.4 Work role	40
2.	Project description	40
	2.1 Core work flow	40
	2.2 Architecture	40
	2.3 Work item	40
	2.4 Project Team	40
3.	Handover plan	40
	3.1 Handover timeline	40
	3.2 Handover method	40
4.	Notice	41
	4.1 Cooperation model	41
	4.2 Frequently Asked Questions	41
5.	Account information	41
	5.1 Site	41
	5.2 Source code	41
	5.3 Resources	41
	5.4 App iTunes/Play Store URL	41
6.	Work Status	41
	6.1 Current Status	41
	6.2 Status project on last day of contract	41

# Project handover plan

## 1. Project background

Digital Inspection application is developed for a b2b car market platform in which inspection data of vehicles from different locations can be saved using a mobile application. Photos of defective parts can be captured and damage can be marked on the image while uploading it to the server.

### 1.1 Client

Jan Mrhar

### 1.2 Purpose

*To manage the vehicles received at the repairing station*

### 1.3 Scope

*Management*

### 1.4 Work role

*Development, Maintenance, Client coordination*

## 2. Project description

### 2.1 Core work flow

*Waterfall model was followed.*

### 9.6 Architecture

*MVVM*

### 2.3 Work item

*Project is completed and delivered.*

### 2.4 Project Team

Tarak Bhawalkar

## 3 Handover plan

### 3.1 Handover timeline

*N.A.*

### 3.2 Handover method

*N.A.*



## **4. Notice**

### **4.1 Cooperation model**

*N.A.*

### **4.2 Frequently Asked Questions**

*N.A.*

## **5. Account information**

### **5.1 Site**

<https://preskok.si/en/home/>

### **5.14 Source code**

*N.A.*

### **5.15 Resources**

*N.A.*

### **5.16 App iTunes/Play Store URL**

*N.A.*

## **6. Work Status**

### **6.1 Current Status**

*Completed*

### **6.2 Status project on last day of contract**

*Completed and Delivered*

**Jawwy TV  
Project handover plan**

**Version <1.0>**

## Release history

<b>Date</b>	<b>Version</b>	<b>Comment</b>	<b>Author (Developer)</b>	<b>Approved By (TL)</b>
N.A.	N.A.	N.A.	N.A.	N.A.

# Index

1.	Project background	45
	1.1 Client	45
	1.2 Purpose	45
	1.3 Scope	45
	1.4 Work role	45
2.	Project description	45
	2.1 Core work flow	45
	2.2 Architecture	45
	2.3 Work item	45
	2.4 Project Team	45
3.	Handover plan	45
	3.1 Handover timeline	45
	3.2 Handover method	45
4.	Notice	46
	4.1 Cooperation model	46
	4.2 Frequently Asked Questions	46
5.	Account information	46
	5.1 Site	46
	5.2 Source code	46
	5.3 Resources	46
	5.4 App iTunes/Play Store URL	46
6.	Work Status	46
	6.1 Current Status	46
	6.2 Status project on last day of contract	46

# Project handover plan

## 1. Project background

Jawwy-TV is an OTT media streaming application designed specifically for Android TV for the audience in the Middle East, in which viewers can watch movies, TV shows, and other media on their android TV.

Based on the content the user watches, they are also shown recommendations on the TV home screen, and if there's some VOD content stopped midway, they can resume the playing from the continue watching section on the home screen itself.

### 1.1 Client

Diagnal

### 1.2 Purpose

*OTT App*

### 1.3 Scope

*OTT App*

### 1.4 Work role

*Development, Client coordination*

## 2. Project description

### 2.1 Core work flow

*N.A.*

### 9.7 Architecture

*MVVM*

### 2.3 Work item

*Project is closed.*

### 2.4 Project Team

Tarak Bhawalkar

## 3 Handover plan

### 3.1 Handover timeline

*N.A.*

### 3.2 Handover method

*N.A.*

## **4. Notice**

### **4.1 Cooperation model**

*N.A.*

### **4.2 Frequently Asked Questions**

*N.A.*

## **5. Account information**

### **5.1 Site**

<https://www.diagnal.com/>

<https://www.jawwy.tv/en/home>

### **5.17 Source code**

*N.A.*

### **5.18 Resources**

*N.A.*

### **5.19 App iTunes/Play Store URL**

*N.A.*

## **6. Work Status**

### **6.1 Current Status**

*Closed*

### **6.2 Status project on last day of contract**

*Closed*