

AI-Powered Digital Signage

This repo is for [P2024-08] AI-Powered Digital Signage for Targeted and Personalized Advertisement.

Introduction

This project aims to develop an AI digital signage capable of collecting users' facial and emotional data via a camera, enabling the recommendation of personalized advertisements tailored to individual preferences, thereby enhancing the fluidity and comfort of the shopping experience for users.

Start up the project

We have provided multiple entrance for user to execute the project

1. For windows users:

Double click the ai_digital_signage_win.exe file and input your camera index(0 is your default camera, 1 is external camera) immediately after the

```
1 | Using device: cpu/cuda
2 | cpu/cuda
3 | Start up. Today's date is: <today's date>
```

prompt shows up. Please reserve at least 5GB of available space on your hard drive. When running for the first time, just after observing that the global tokenizer is loaded to 100%, immediately enter 0 and press Enter and wait for the program to run.

2. For Mac users:

Mac users should download our ai_digital_signage_mac executable file; however, it is typically recognized as a document. In this case, please refrain from opening it. Instead, in the current directory, execute the following command in the terminal:

```
1 | chmod +x ai_digital_signage_mac
```

This command will extract it as a executable file, then users can execute it by clicking or using the commands below in the terminal:

```
1 | /dist/ai_digital_signage_mac; exit;
```

If a security warning is triggered and you are unable to open the file, please go to the settings and trust the file under "Privacy & Security".

3. To run the source code

Before get started, you may need to install PyTorch on your device:<https://pytorch.org/>.
Upgrade pip with:

```
1 | pip install --upgrade pip
2 | pip install --upgrade setuptools
```

Install transformer package with:

```
1 | pip install transformers
```

Update related dependencies with:

```
1 | pip install --upgrade torch pyinstaller onnxscript dash
```

If any problem occurs while installing dash:

```
1 | pip install setuptools wheel
2 | pip install dash --no-binary stringcase
```

You may need to install the required dependencies, execute the following command to install them in [requirements.txt](#).

```
1 | pip install -r requirements.txt
```

Now you could run the program with:

```
1 | python state.py
```

A token for the LLM model Llama-3.2-1B-Instruct is required, we provide you an available Hugging Face token in the .env file.

You may need VPN to login the Hugging Face for the first time.

The default camera is assigned the index 0. If you have an external camera, please enter 1, you can enter the index according to your requirement.

```
1 | python state.py
2 | Using device: cpu
3 | cpu
4 | Start up. Today's date is: 2025-03-24
5 | Input camera index:
```

It will open a port to display the frontend components of the project automatically. In order to get a better using experience, we strongly recommand you to use full screen mode.

The system acquires access to the camera resource, it will immediately initiate face detection and retrieve facial and emotional data.

```
1 | 0: 384x640 1 face, 303.1ms
2 | Speed: 16.6ms preprocess, 303.1ms inference, 22.1ms postprocess per image at shape (1,
3 | 3, 384, 640)
4 | [CV] Predicted Demographics: ('17-35', 'Male', 'Asian', 'sad')
5 | [CV] Putting to detected_face_queue...
6 | [State] LLM Processing: Generating ad text.
```

Three browser windows will show up as below:

- Main screen will display advertisements and show to the users.
- Secondary screen shows the demographic data, emotion data and user's image.
- Dashboard is to show the watching data for the stakeholder. The watch time is recorded only when the user focuses on the main screen. You may click refresh button to update the newly arrived data on the dashboard.

Main Screen

- Ad rotating state: randomly display advertisements

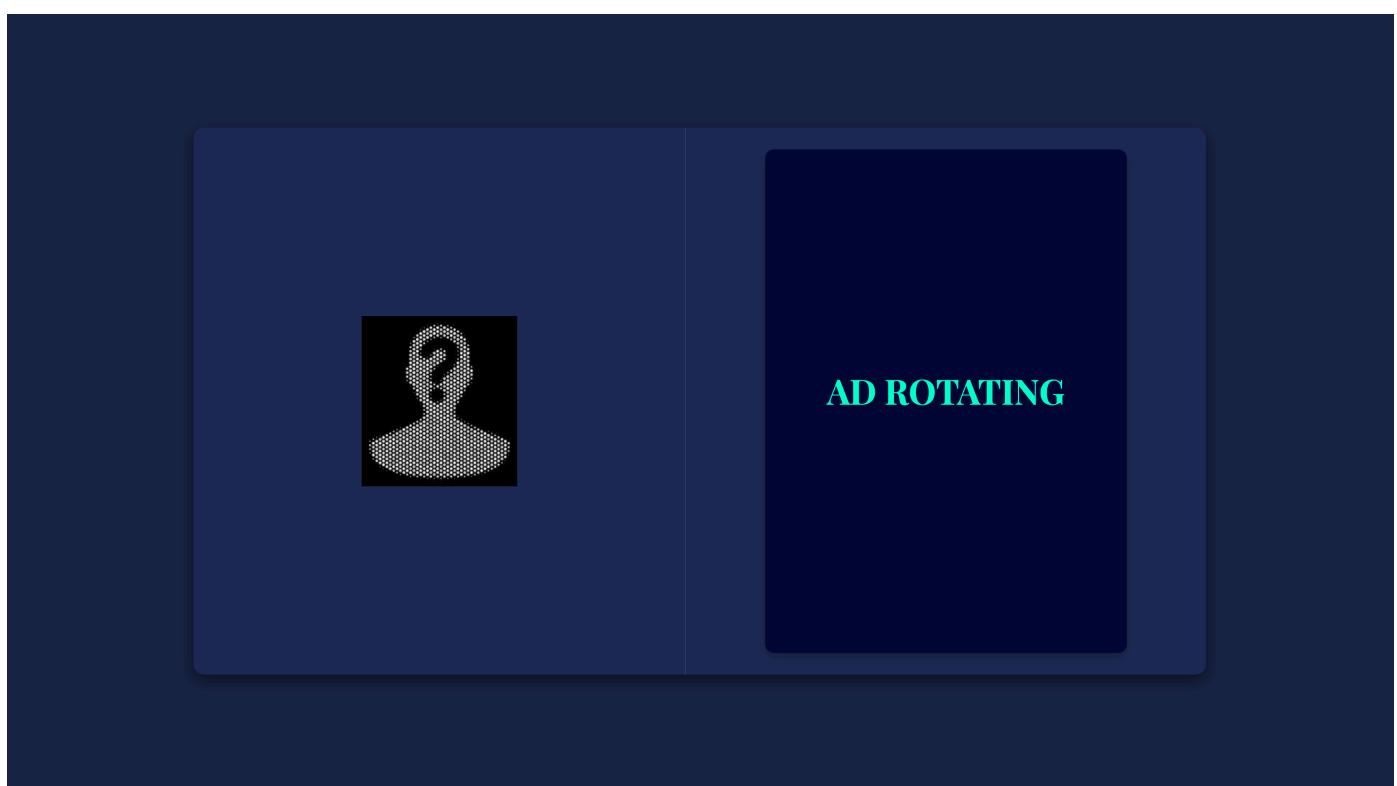


- Targeted video displaying state: display personalized advertisements according to the demographic data and emotional data

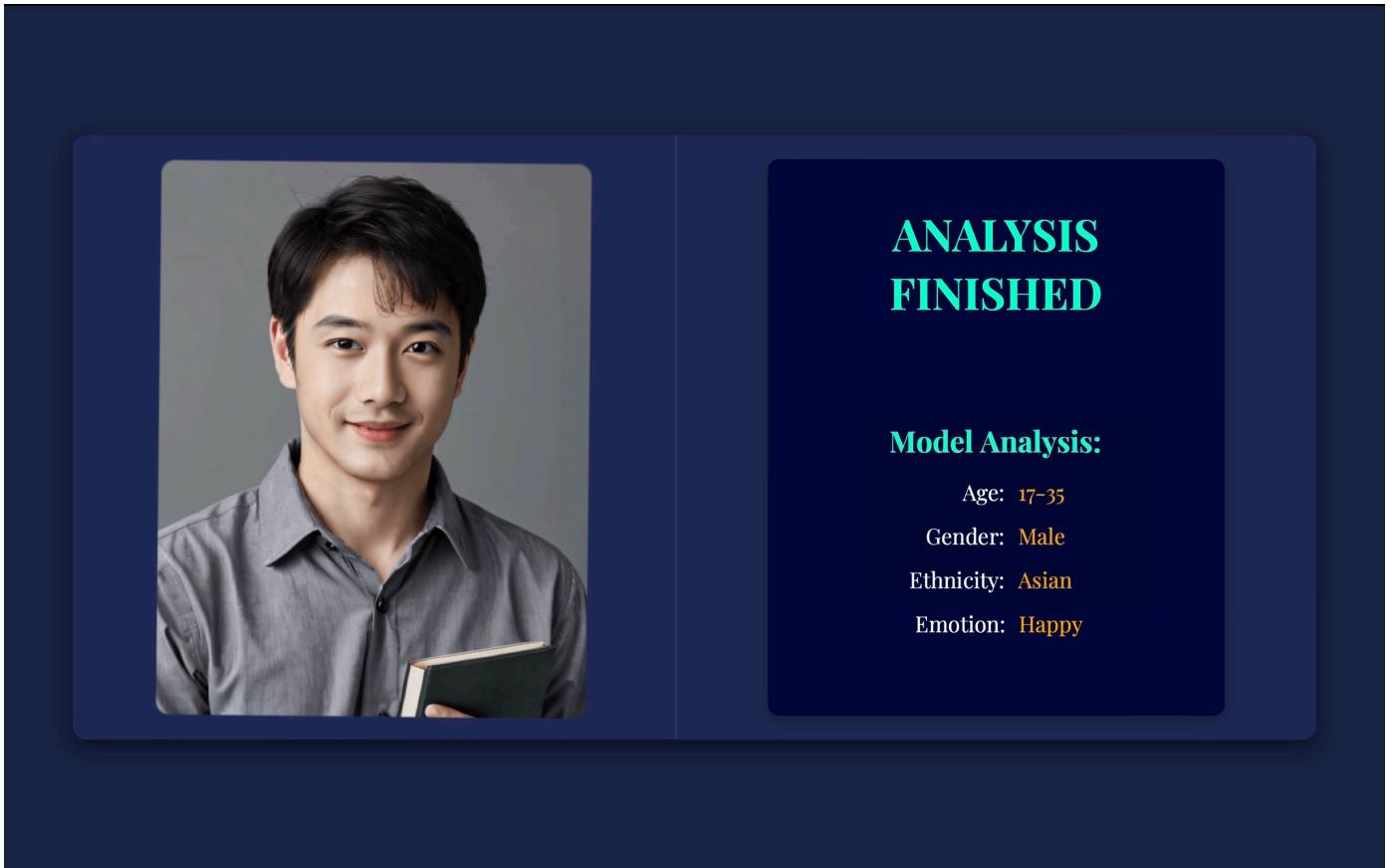


Secondary screen

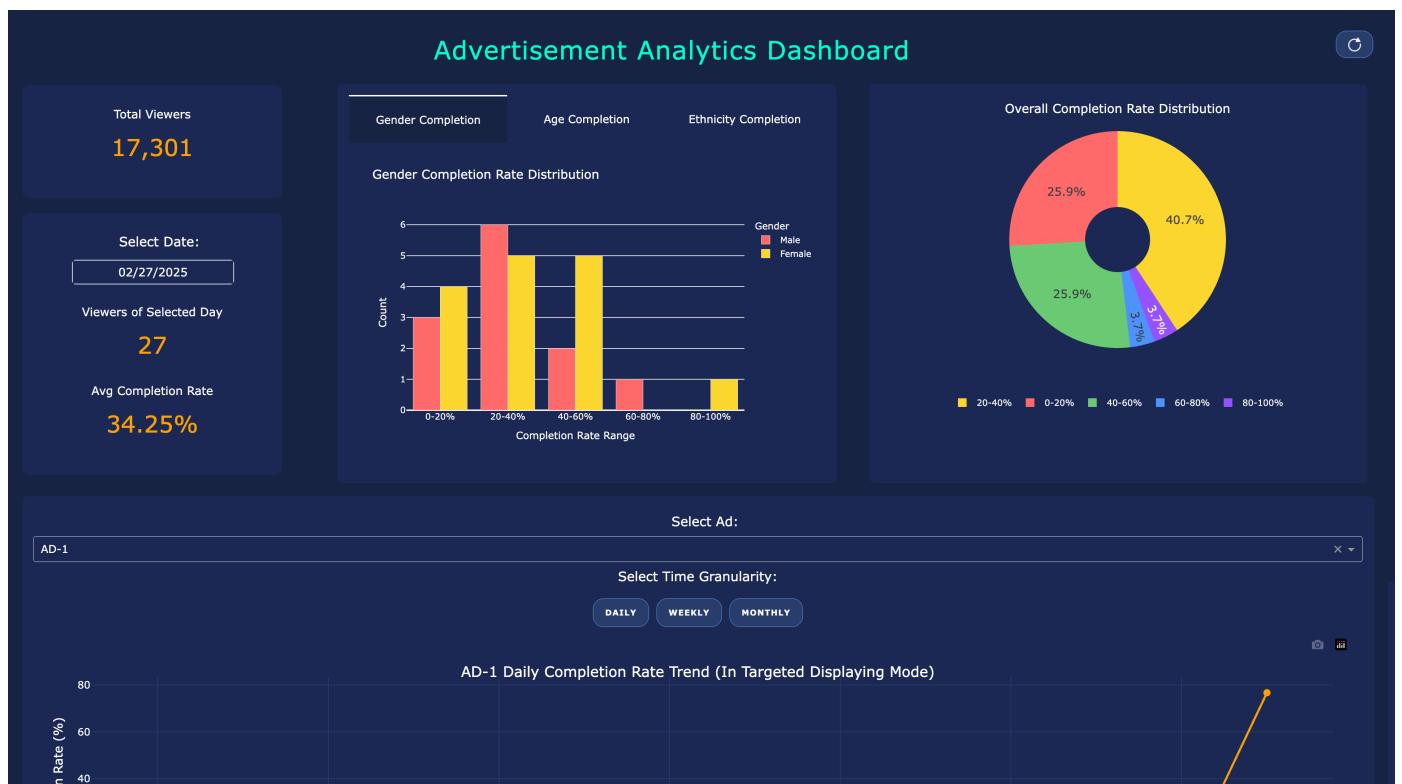
- Ad rotating state: no face detected



- Targeted video displaying state: display demographic data and emotional data analyze result



Dashboard



If you have any confusion about the project, please contact us:

scyal8@nottingham.edu.cn

Quality Assurance

The system contains four main Modules, they are:

Module ID	Module Name	Description
M1	CV Module	Computer vision module for image processing and visual analysis
M2	LLM Module	Large Language Model interface for content generation
M3	Display Module	Front-end video rendering and UI presentation
M4	Analytical Module	Data processing and business intelligence dashboard

The functional requirements are identified as:

RID	Requirement	DID	Design Description	Module ID
FR1	The user can engage with the digital signage to view personalized advertisement videos.	D1.1.1	LLM Interface	M2
		D1.1.2	Front-end Page	M3
		D1.1.3	Database	M4
FR2	Users can pass completion data to the dashboard through the system	D1.2.1	Eye-tracking Interface	M1
		D1.2.2	Dash analytics	M4

- RID: Requirement ID
- FR: Functional Requirement

The non-functional requirements are identified as:

RID	Requirement	DID	Design Description	Module ID
NFR1	The system should detect and identify user emotions and demographic information.	D2.1.1 D2.1.2	Real-time face detection Emotion recognition pipeline	M1
NFR2	The system should get the response from LLM within 15 seconds.	D2.2.1 D2.2.2	Local deployment Prompt engineering	M2
NFR3	The system should use the database to save the relevant information of the advertising pool and user completion.	D2.3.1 D2.3.2	Database storage Ad matching	M4
NFR4	The system should use the integration of eye-tracking models to correctly obtain the ad completion.	D2.4.1 D2.4.2 D2.4.3	Eye-tracking model integration Engagement metrics DB Dashboard display	M1 M4 M4
NFR5	The system should complete demographic analysis in 0.5s	D2.5.1 D2.5.2	Face Detection Frame Analysis	M1
NFR6	The system should obtain completion based on user eye movement data.	D2.6.1 D2.6.2	Eye-tracking Model Metrics storage	M1 M4
NFR7	The system should not retain user privacy data. (GDPR)	D2.7.1 D2.7.2	Ephemeral storage Instant analysis	M4 M1
NFR8	The system should complete the state machine function smoothly.	D2.8.1 D2.8.2	Async processing GPU acceleration	M1,M2 M1,M2
NFR9	The system should correctly display and complete the logic of the user interface.	D2.9.1 D2.9.2	Dual-screen layout Dynamic text overlay	M3
NFR10	The codebase should meet key software engineering development standards for later maintenance.	D2.10.1	Modular architecture	All
NFR11	The system should run the state machine logic normally.	D2.11.1	State machine logic	All

- RID: Requirement ID
- NFR: Non-functional Requirement

The team conducted Unit Tests, they are:

CV Tests							
TID	Scenario	RID	Test File	Test Input	Expected Output	Test Output	Test Result
1	Accuracy test	NFR1, NFR5	Test/CV/yo lov8_test.py		(17-35, Male, Asian, Happy) in 0.5s	(17-35, Male, Asian, Happy) in 0.5s	Pass
2					(17-35, Male, Asian, Neutral) in 0.5s	(17-35, Female, Black, Neutral) in 0.5s	Pass
3					(17-35, Female, Asian, Happy) in 0.5s	(17-35, Female, White, Happy) in 0.5s	Pass
4					(17-35, Female, Asian, Sad) in 0.5s	(35-50, Male, White, Angry) in 0.5s	Pass
5					(17-35, Male, Asian, Angry) in 0.5s	(50+, Female, Asian, Happy) in 0.5s	Pass

LLM Test							
TID	Scenario	RID	Test File	Test Input	Expected Output	Test Output	Test Result
6	Functional test	FR1, NFR2	Test/LLM /llm_test.py	('17-35', 'Female', 'Asian', 'happy'); videos/0041.mp4(High heels)	targeted advertisement message according to test input	"Hi high heels are more than just shoes for young women - they're a symbol of confidence, elegance, and personal style."	Pass
7				('35-50', 'Male', 'White', 'sad'); videos/0012.mp4(suits)	targeted advertisement message according to test input	"A well-fitted suit embodies comfort & professionalism; perfect for formal events, business meetings, family gatherings & special occasions."	Pass
8				('35-50', 'Female', 'Black', 'sad'); videos/0028.mp4(lipsticks)	targeted advertisement message according to test input	"A well-fitted suit embodies comfort & professionalism; perfect for formal events, business meetings, family gatherings & special occasions."	Pass

Eyetracking test							
TID	Scenario	RID	Test File	Test Input	Expected Output	Test Output	Test Result
9	Accuracy Test	FR2, NFR4, NFR6	Test/eyetrack/eye_tracing_test.py	Watched For 5.35s	5.35	4.96	Pass
10				Watched For 14.67s	14.67	14.2	Pass
11				Watched For 10.88s	10.88	11.23	Pass
12				Watched For 9.44s	9.44	8.64	Pass
13				Watched For 13.16s	13.16	13.47	Pass

DashBoard Test							
TID	Scenario	RID	File	Test Input	Expected Output	Test Output	Test Result
14	Functional test	NFR4	Test/Dashboard/dashboard.py	(03/03/2025, AD-1, gender Completion rate, daily)	The dashboard would select data of 03/03/2025 from database, the bar chart would show the gender completion rate and line chart below will show the ad-1 completion rate daily	update as expected	Pass
15				(25/02/2025, AD-53, ethnicity Completion rate, weekly)	The dashboard would select data of 25/02/2025 from database, the bar chart would show the ethnicity completion rate and line chart below will show the ad-53 completion rate weekly	update as expected	Pass
16				(02/02/2025, AD-23, gender Completion rate, monthly)	The dashboard would select data of 02/02/2025 from database, the bar chart would show the gender completion rate and line chart below will show the ad-23 completion rate monthly	update as expected	Pass
17				(18/02/2025, AD-63, age Completion rate, monthly)	The dashboard would select data of 02/02/2025 from database, the bar chart would show the gender completion rate and line chart below will show the ad-24 completion rate monthly	update as expected	Pass
18				(22/02/2025, AD-14, ethnicity Completion rate, daily)	The dashboard would select data of 22/02/2025 from database, the bar chart would show the ethnicity completion rate and line chart below will show the ad-14 completion rate daily	update as expected	Pass

The team conducted Integration and System Tests, they are:

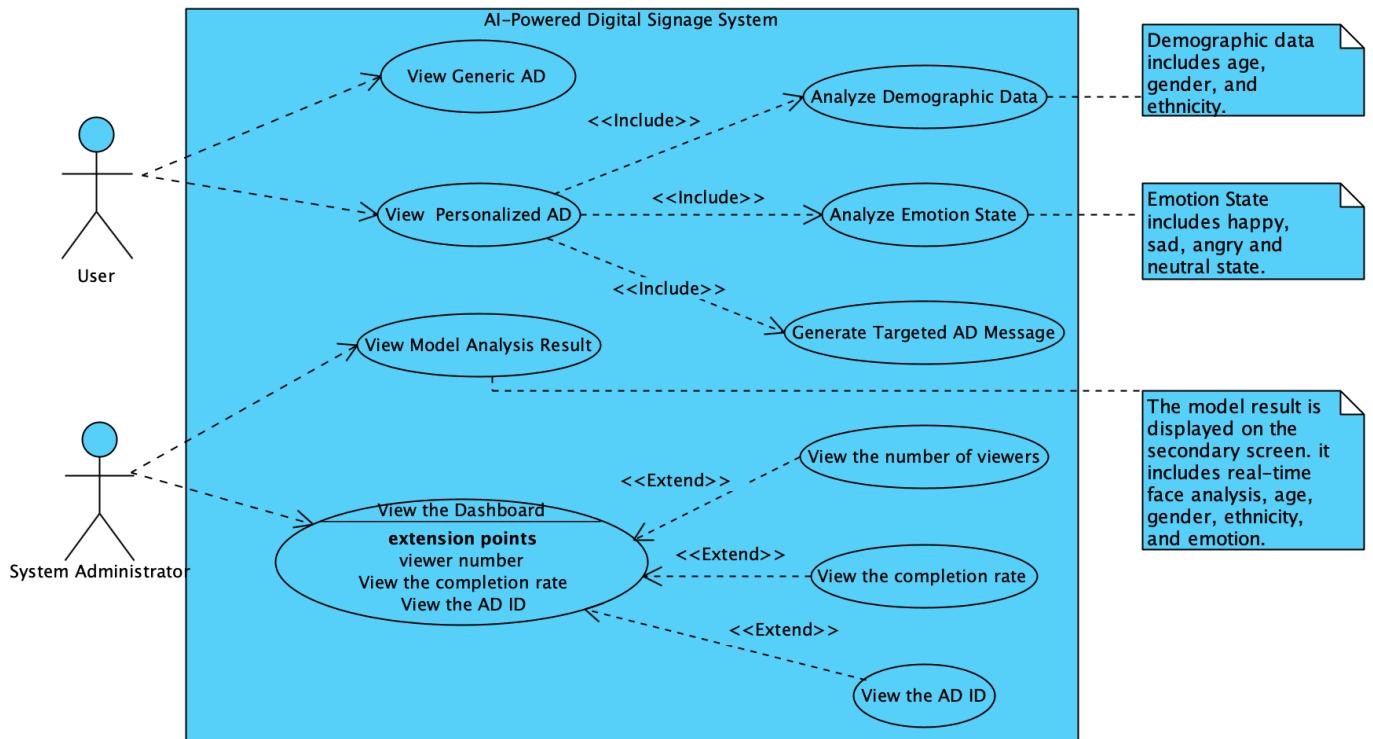
TID	Scenario	RID	Test File	Input	LLM Test		Test Output	Test Result
					Expected Output			
19	Functional test	FR1, NFR2	Test/LLM/llm_test.py;Test/CV/yolov8_test.py		('50+', 'male', 'White', 'angry'); videos/0046.mp4(Organic Vegetables) targeted advertisement message according to test input	"Indulge in our organic vegetables grown using natural farming methods without synthetic pesticides, fertilizers & GMs. Fresh, high nutritionally complete, reduced exposure to toxins, promoting peace of mind."	Pass	Pass
20					('50+', 'Female', 'White', 'sad'); videos/0008.mp4(Sunglasses) targeted advertisement message according to test input	"Feeling down? Our stylish sunglasses protect your eyes from harsh UV rays, reduce glare & improve visibility with clear comfort & bright colors."		

State Test---Secondary Screen								
TID	Scenario	RID	Test File	Test Input	Expected Output	Test Output	Test Result	
26	Logical Test	NFR11	Test/state_test.py	No Input	Stay in ad rotating screen	Output as expected	Pass	
27				('17-35', 'Female', 'Asian', 'happy')	received CV data, show user's photo, age gender race and emotion	Output as expected	Pass	
28				('35-50', 'Male', 'White', 'sad') then no input	received CV data, show user's photo, age gender race and emotion then return to ad rotating screen	Output as expected	Pass	
29				('35-50', 'Female', 'Black', 'sad') then ('50+', 'Female', 'Other', 'angry')	received CV data, show user's photo, age gender race and emotion then return to ad rotating screen and then back to received CV data state	Output as expected	Pass	
30				('50+', 'Male', 'Indian', 'sad')	received CV data, show user's photo, age gender race and emotion	Output as expected	Pass	

Hence, the team ensured all requirements have been testified and satisfied.

Appendix

Use Case Diagram For The System



License

Click here to see [Licenses](#).