

# TERNOPS

## Customer Meeting 3

# Agenda

Project Overview

Demo

Tests

Build Script

Documentation Plan

Q & A

# 01

## Project Overview

# Iteration 1

- Starting on the add/edit/delete functionality.
- Creating a basic GUI framework.
- Log in / log out.



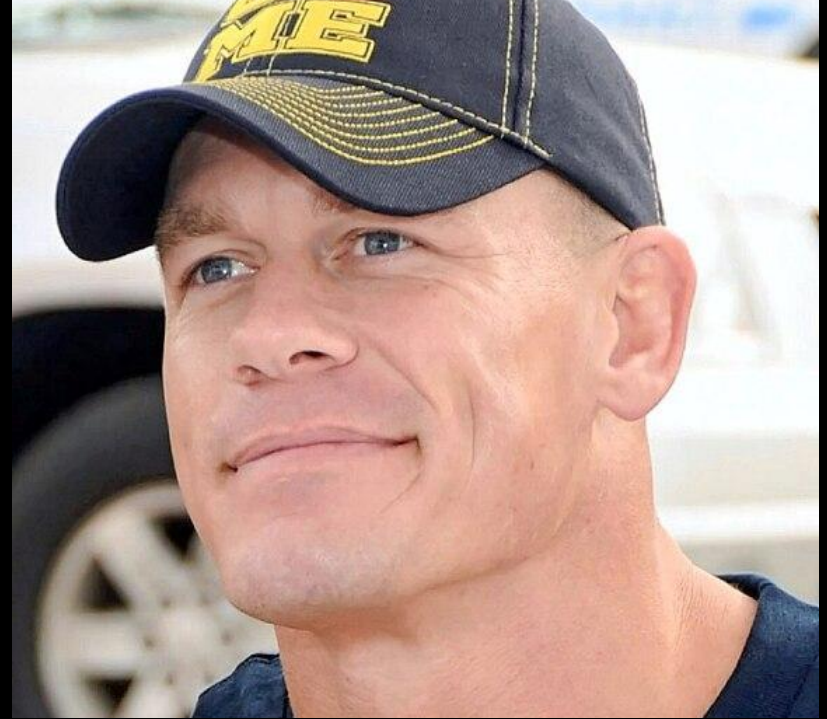
# Iteration 2

- Browse celebrities.
- Search bar, to find by name.
- Filter celebrity profiles by awards, etc.
- Button for log in/out & sign up.
- Homepage for seeing trending celebrities.
  - When user logged in they should be able to see the favourites list.



# Iteration 3

- View image(s) of celebrity.
- View video(s) of celebrity.
- Hierarchical system of permission for add, edit & delete functions.
- Favoriting celebrities that makes a list of them for easy find when re-logging in.



Done

As a user, I want a homepage so I can see trending celebrities and all of my favourites (if I am logged in)

As a user, I want to be able to log in, so that I have access to more features of the catalog, like favouriting.

As a user, I want to search celebrities by name, so that I can quickly access the info I need for my research.

As a user, I want to be able to delete information if it is irrelevant or incorrect, so the information is kept recent and correct.

As a user, I want to be able to edit the existing information on celebrities, so that the biography of the celebrity is accurate and up to date.

As a user, I want to be able to add new celebrities to the database so the database keeps up with growth of industries and emergence of new famous people.

As a user, I want to have a user friendly interface so I can easily interact with the application and see the information on a visually appealing medium.

As a user I want a dedicated button to create an account/log in/sign out so I can log in/out with ease.

+ Add a card

Iteration 1

As a user, I want to be able to log in, so that I have access to more features of the catalog, like favouriting.

As a user, I want to be able to delete information if it is irrelevant or incorrect, so the information is kept recent and correct.

As a user, I want to be able to edit the existing information on celebrities, so that the biography of the celebrity is accurate and up to date.

As a user, I want to be able to add new celebrities to the database so the database keeps up with growth of industries and emergence of new famous people.

As a user, I want to be able to filter the celebrity profiles by awards won, industry, etc. so that I can specifically find what I want to find instead of browsing aimlessly.

As a user, I want to have a user friendly interface so I can easily interact with the application and see the information on a visually appealing medium.

+ Add a card

Iteration 2

As a user I want a dedicated button to create an account/log in/sign out so I can log in/out with ease.

As a user, I want to be able to filter the celebrity profiles by awards won, industry, etc. so that I can specifically find what I want to find instead of browsing aimlessly.

As a user, I want to search celebrities by name, so that I can quickly access the info I need for my research.

As a user, I want a homepage so I can see trending celebrities and all of my favourites (if I am logged in)

+ Add a card

Iteration 3

As a user, I want to be able to see image(s) of the celebrity I'm looking for so that I have a better idea of what the celebrity looks like.

As a user, I want to be able to see video(s) of the celebrity I'm looking for so that I can hear their voice, and see them in action.

As a user, I want the product to have a hierarchal system so that some users have more permissions than others, so the general public can't add misinformation.

As a user, I want to be able to favourite celebrities, so that when their page is updated I can quickly find the page for my favourite celebrities.

+ Add a card

Backlog

As a user, I want to be able to see video(s) of the celebrity I'm looking for so that I can hear their voice, and see them in action.

+ Add a card

In Testing

As a user, I want to be able to filter the celebrity profiles by awards won, industry, etc. so that I can specifically find what I want to find instead of browsing aimlessly.

As a user, I want to browse celebrities, so that I can find info about celebrities I'm interested in.

As a user, I want to be able to see image(s) of the celebrity I'm looking for so that I have a better idea of what the celebrity looks like.

+ Add a card

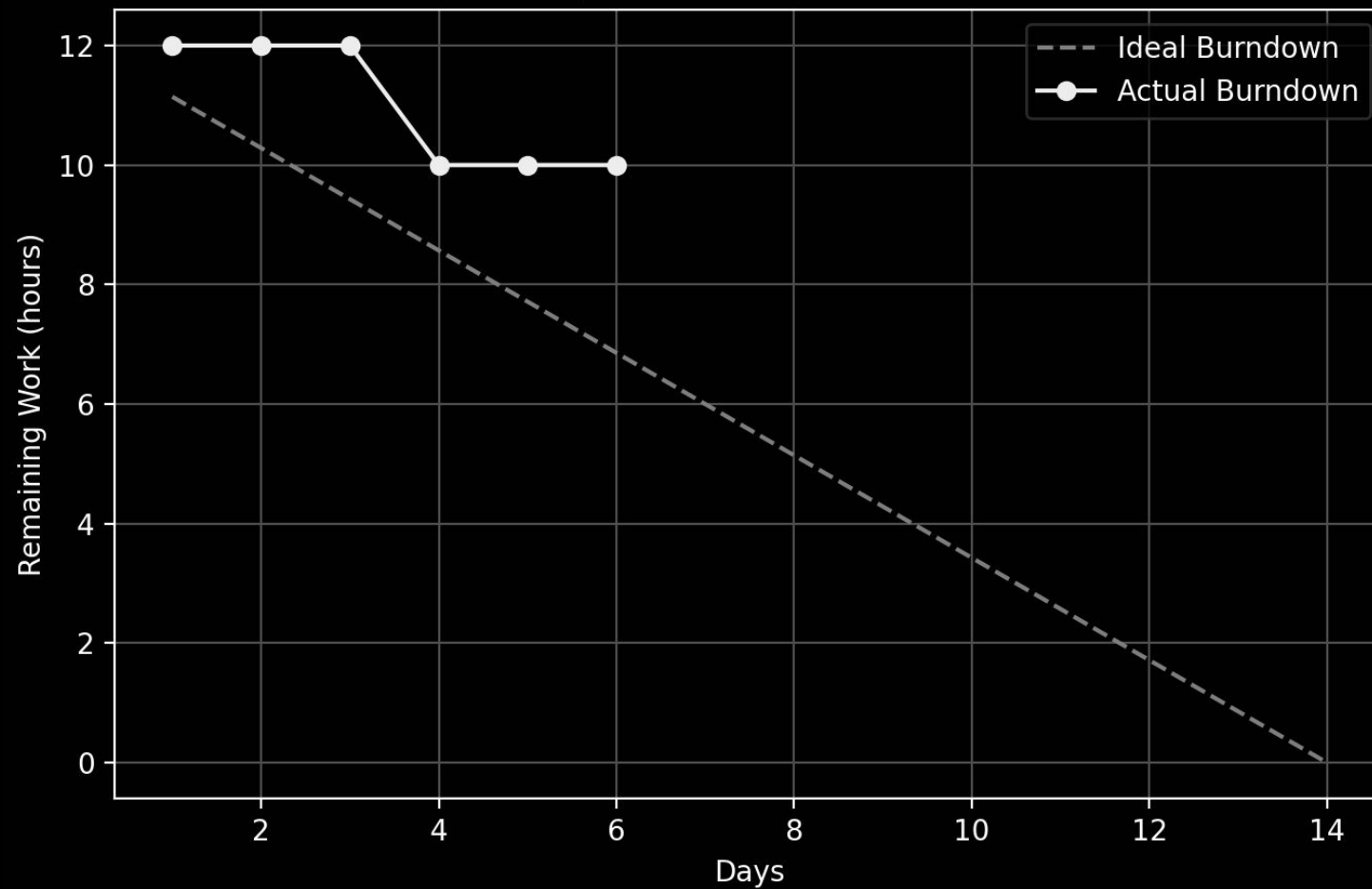
In Progress

As a user, I want the product to have a hierarchal system so that some users have more permissions than others, so the general public can't add misinformation.

As a user, I want to be able to favourite celebrities, so that when their page is updated I can quickly find the page for my favourite celebrities.

+ Add a card

# Sprint Burndown Chart





# 02

## Demonstration

# 03

## Tests

# Testing Plan

- We have created an in-depth testing suite that tests all major functionalities of the program.
- We automated many of these tests using Python's unittest library to efficiently and accurately test the program and every component in it.
- We used CB testing for testing account-related functionality, TB testing for any functions that deal with our database (add, edit, delete), and OB for testing functions where we are looking for correct outputs within the system.
- We test edge cases and unexpected use cases.



# Unit Tests

Test ID	Method/Class	Input(s)	Expected Output(s)	Testing Approach	Assigned Team Member
UT-01-CB	register_user()	"John" "abc123"	"Registered"	Automated (unittest)	Tomasz
UT-02-CB	register_user()	"John" "abc123"	"Sorry, the username John is already taken, Please try again."	Automated (unittest)	Tomasz
UT-03-CB	login()	"John" "abc123"	"Logged In"	Automated (unittest)	Tomasz
UT-04-CB	login()	"Bill" "abc123"	"No account found for the username Bill"	Automated (unittest)	Tomasz
UT-05-CB	login()	"John" "123abc"	"Incorrect password"	Automated (unittest)	Tomasz
UT-06-CB	log_out()	null	"Successfully logged out"	Automated (unittest)	Tomasz
UT-07-TB	add_data()	"Eric" "Rapos" "July 20, 1980" "eric_rapos.jpg"	"Successfully added a new entry for Eric Rapos."	Automated (unittest)	Tomasz
UT-08-TB	add_data()	"Eric" "Rapos" "July 20, 1980" "eric_rapos.jpg"	"Entry for Eric Rapos already exists."	Automated (unittest)	Tomasz
UT-09-TB	edit_data()	"first_name" "Eric" "date_of_birth" "July 20, 1970"	"Successfully updated Eric Rapos for "	Automated (unittest)	Tomasz
UT-10-TB	delete_data()	"Eric" "Rapos"	"Successfully deleted the entry for Eric Rapos."	Automated (unittest)	Tomasz

# Integration Tests

Test ID	Components Involved	Preconditions	Steps	Expected Result	Assigned Team Member
IT-01-CB	searchCelebrities() & filterCelebrities() & GUI & CSV file	GUI initialized & celebrity csv is loaded & User can interact with search bar	Enter name into search bar, submit	Search successfully filters and is displayed	Nolan
IT-02-TB	createEditCelebrityFrame("Add") & createCelebrityRow() & GUI	GUI initialized & celebrity csv loaded & User can press "+" button	Click the "+" button, enter details and save	Addition is successful and appears on display, remains in csv after closure	Nolan
IT-03-TB	createEditCelebrityFrame("Edit") & createCelebrityRow() & GUI & CSV file	GUI initialized & celebrity csv loaded & User can press the "Edit" button	Click the "Edit" button, edit the details and save	Editing is successful. It appears on display and is saved in csv	Nolan
IT-04-CB	createFilterFrame() & filterCelebrities() & GUI	GUI initialized & celebrity csv loaded & User can press filter dropdown	Click the filter dropdown and select filters and press search	Filtering is successful. Displays results based on filter.	Nolan
IT-05-CB	createCelebrityFrame() & GUI	Initialized GUI & Celebrity csv loaded & User can press learn more	Press the "Learn More" button.	Successfully displays all information about celebrity	Nolan
IT-06-TB	removeCelebrity() & GUI & CSV file	Initialized GUI & Celebrity csv loaded & User can press delete	Press the "Delete button" and confirm the delete	Successfully deleted from csv and no longer is displayed	Nolan

# System Tests

Test ID	Scenario	Preconditions	Steps	Expected Outcome	Assigned Team Member
ST-01-TB	Adding a new celebrity to the system	App is running, user can press the "+" button	User presses the "+" button, then they fill in at least the Full name and industry of the celebrity, and press the submit celebrity button	The new celebrity is added to the CSV and is visible in the app's main view. A success message is shown.	Matteo
ST-02-TB	Editing an existing celebrity's data	App is running, user can press the "edit" button, there is an existing celebrity	User presses the "edit" button, then they edit what the need to in the edit celebrity form, and press the edit celebrity button	The celebrity's updated data is saved in the CSV, and the changes are reflected in the app	Matteo
ST-03-CB	Filtering celebrities based on search criteria	App is running, user can press the "filter" button, there are celebrities	User presses the "filter" button, they then select filter criteria (date of birth, genre, etc.)	The display should be a filtered list of celebrities that match the selected criteria	Matteo
ST-04-OB	Displaying detailed celebrity info	App is running, user can press the "filter" button, there is a celebrity	User selects a celebrity from the homepage screen, then they click the "Learn More" button.	The celebrity's full details (biography, achievements, etc.) display in the UI	Matteo

# 04

## Build Script

# Build Script

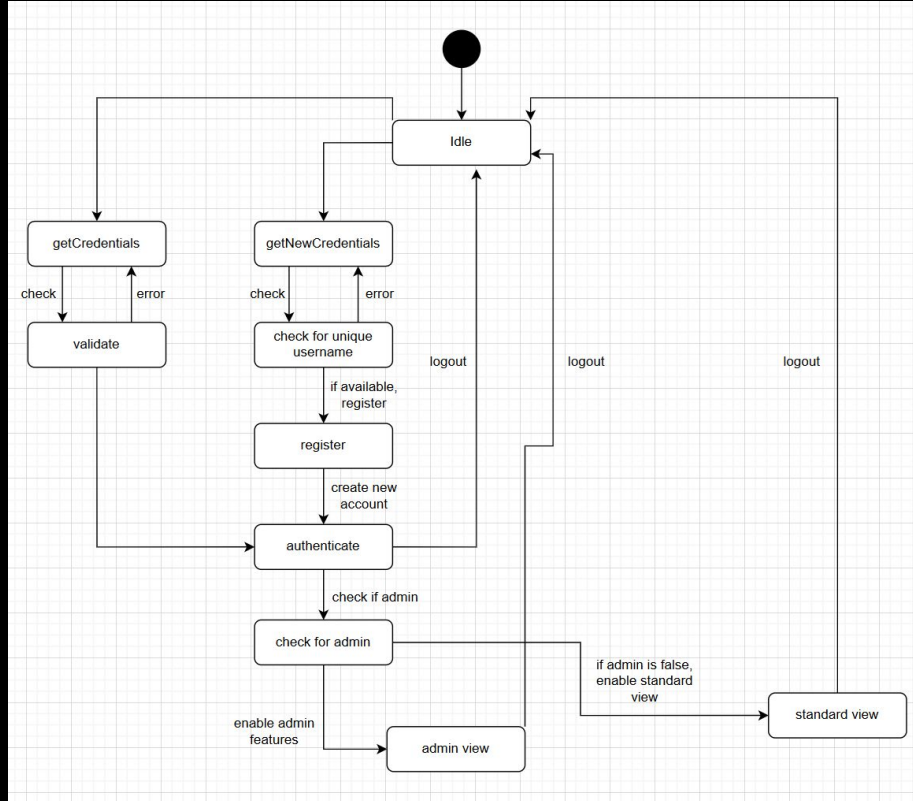
```
1  #!/bin/bash
2
3  # Exit immediately if a command exits with a non-zero status
4  set -e
5
6  # Step 1: Check if Python3 is installed
7  if ! command -v python &>/dev/null; then
8      echo "Python3 is required but not installed. Please install Python3."
9      exit 1
10 fi
11
12 # Step 2: Check if pip is installed
13 if ! command -v pip &>/dev/null; then
14     echo "pip is required but not installed. Please install pip."
15     exit 1
16 fi
17
18 # Step 3: Set up the virtual environment
19 echo "Setting up virtual environment..."
20 python -m venv venv
21
22 # Step 4: Activate the virtual environment
23 echo "Activating virtual environment..."
24 source venv/Scripts/activate
25
26 # Step 5: Install dependencies
27 echo "Installing dependencies from requirements.txt..."
28 pip install -r requirements.txt
29
30 # Step 6: Run the GUI application
31 echo "Running the application..."
32 python src/gui.py
33
34 # Step 7: Deactivate the virtual environment after running
35 deactivate
```



# 05

## Documentation Plan

# Documentation Plan



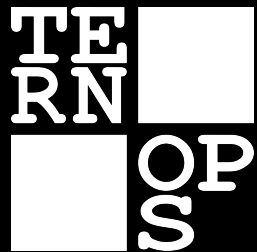
Assigned tasks:

1. Developer Documentation
  - a. Code Documentation
    - i. In the python files → *Matteo*
  - b. UML / Class Diagram Documentation
    - i. A pdf document containing diagrams created throughout the project → *Erica*
2. User/In application documentation
  - a. User Manual
    - i. The README.MD in the github → *Nolan*

# 06

## Q & A

\*any final features you REQUIRE by the time of the last customer meeting?



**Thank You**