

**Amirkabir University of Technology
(Tehran Polytechnic)**

StoryPal Application

Software Requirements

Team Members

Anita Seihoon (Team Mentor)

Parsa Saei 9812028

Amir Shahbazi 9812033

AmirHossein Rostami 9813016

Mohammad Amin Ghasemi 9813067

What is StoryPal?	4
Our Stakeholders	6
Shareholders	6
Writers (Authors, Game Developers, Script Writers, etc)	6
Artists	7
Readers	7
Publishers	7
Application Requirements Specification	9
Asset Creation	9
Map Creation	9
Character Relations	9
TimeLine	9
Script Writing	10
Online Interaction	10
Security	10
Quality Function Deployment	11
Expected Requirements:	11
Normal Requirements:	12
Exciting Requirements:	12
Channels and Budget	14
Links:	14
Use cases	16
Use case 1: Creation Process	16
Use case 2: Deployment Process	18
Use case 3: Reader Experience	20
Use case 4: Publisher Experience	22
Software Requirements Specification (SRS)	24
Front end	24
Back end	24
Figma MVP	26
Variants	27
Complex Components	28
Introductory Page	29
Login Fail	30
Create New Account	31

Create New Account Example	32
Sign-up Verification	33
Verification Failed	34
Forgot My Password	35
Set New Password	37
Home Page	38
Side-bar Menu	39
User Profile	40
Create New Project	41
Open Project	43
Delete Project	44
Project Panel	45
Asset Page	46
Asset Creation	47
Story Map	48
TimeLine	49
Character Relations	50
Story Script	51
State Diagram	52
Structural Diagrams	55
Data Flow Diagram (DFD)	55
Class Diagram	58
Component Diagram	59
Deployment Diagram	60



"Having something to say is the easy part. But moulding and delivering that message, carving it in continuity and pushing every piece alongside it while remembering where and why every element coexists, is where the memory, effort, time and struggle come in.

It's where the talent comes in. It's what separates the great."

Storytelling is one of the few human acts that transcend culture, transcend History; Help to spread a message; Teach a lesson; Motivate people to better themselves; Or simply offer an escape to worlds we can only dream of.



Yet over the years, our standards for good storytelling have dropped. Many stories are written without care; They lack consistency within themselves; payoffs that don't have set-ups, etc; They can have unforeseen consequences that go against the message; Many are concerned with having something to say rather than saying it right.

And we see many people new to storytelling that cannot put their ideas on the page, that lack the experience or the motivation to keep going with their story. They might see it as a cloud of random thoughts without structure. Or maybe they write their work but cannot bring it to attention. They become unknown authors that have the talent to create great works, but not the support to publish and spread them.

What is StoryPal?

In a 2014 survey taken from "Digital Book World and Writer's Digest Author Surveys" over 9000 respondents were questioned, and of all the people who completed a manuscript for their project, only about 23% succeeded to be traditionally published.

Since the start of the Covid pandemic, many have been spending their time in their homes. And for some, this has been an opportunity to start writing. The number of new writers has gone up significantly and it is likely to keep increasing.

The most popular writing apps have seen an increase in their user base in the last 2 years. Yet they still have major flaws within their app. "Campfire writing" has reached over 100,000 users from only 30,000 in the past 2 years and "WorldAnvil" has risen to the top 5 writing apps thanks to the surge of new users.

They offer many features for users, creating characters, making maps, writing tutorials, etc. but we know that writing is more advanced than just writing character descriptions, it involves development, changes throughout the story to the plot, world and its inhabitants; it can be in the form of the location of an object; it can be the development a character experiences over their journey, etc. hence making these features so that they can dynamically evolve in sync with the story should be a foundation, which all major writing



apps lack. This is the major hurdle to pass to compete with the standards of writing apps.

We plan to take a new look at how writing apps should function, acting more dynamically; Feature parameters change based on the story and its development. Helping the user by providing a default skeleton to work with while having the option to change it if they prefer.

StoryPal is an application that assists writers to create a well-written story. It aims to help them structure their ideas and easily bring them to the page. and when a story is written, it helps the writer connect with a community of people to receive feedback, criticism, support and advice to better their craft. And in the way, gather an interested and active fanbase.

We aim to achieve that in two stages: Creation and Deployment.

In the Creation section writers can create a project, make world maps, create characters with their attributes, visualise and draw the relation between these characters, and most importantly, use the Timeline feature to develop these aspects based on the story progression.

In the Deployment section writers can upload their project there it can be accessed by other users to some degree, making other writers able to download certain parts of the project to help their own story (like a map), allowing the readers to give feedback to the writer while being up-to-date on the story's creation progress. It can also be read by editors and reviewers, which can result in either consulting over the editing process or maybe a deal to publish the book.

Our Stakeholders

Based on Sommerville and Sawyer [Som97], a stakeholder is defined as "Anyone who benefits directly or indirectly from the system which is being developed."

We can identify anyone who directly benefits from our application to be internal stakeholders, they can have shares in our business, be employed by our company, etc.

We can then also define external stakeholders as those who do not directly work with our company but are affected by our app.

With this in mind, we can then list our main stakeholders which will form the majority of our application requirements based on their needs.

Shareholders

App developers, designers, investors, etc, benefit directly from the app's success in the market, so they want a popular app with a well-thought-out monetization strategy.

Writers (Authors, Game Developers, Script Writers, etc)

Writers will be the majority of our user base and they will interact with most of our app features. the main goal of the app is to help writers create and share their stories with others; Like:

A comprehensive system that structures their story ideas.

A place to neatly store all the information of the story.

An easy to work system that they can write for long periods.

The tools to visualise more complicated parts of writing.

A community to share their story, get useful assets for their story and receive constructive criticism and support.

A place where they can more easily contact publishers and editors.

Artists

The app provides opportunities for Map and asset creation, many of which will be useful to writers for their stories. So a provider of these assets will be artists. They can create detailed world maps, make fantasy creature assets, sci-fi crafts, etc., which will be useful to writers.

Readers

An easy way to interact with writers and stories, the ability to give feedback to writers, and the ability to follow a story's creation and its development. A way to find stories from lesser-known writers.

Publishers

An easy way to discover writers and contact them for publishing.

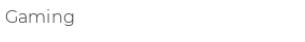
Ability to consult writers with story editing.

User interaction is of utmost importance in our Application. That is why having a great understanding of our target user base and what they expect matters a great deal as well.

So we created a questionnaire to know our potential market and how they regularly interact with stories and gauge the average user's expectations and what they want in a writing app. Then we asked people to join and answer these questions in suitable communities in writing and media criticism influence circles (mainly writing subreddits, media criticism discord channels and D&D Groups on social media).

The questionnaire link is provided below and based on an analysis of the results, a persona is made for the average user of our app.

<https://forms.gle/URmYaPbxTuEzqYDu7>

NAME Mewbshlie Longman		TYPE Rational
 <div> Frustrations <ul style="list-style-type: none"> • Writer's Block. • Stories not making sense. • Constant distractions while writing. </div> <div> Expectations <ul style="list-style-type: none"> Wants an easy to navigate UI. Expects to be able to work on multiple projects at the same time. Wants to Comment on their friends' projects. Wants the option of sharing his story. Wants an in-depth Character Creation. Wants the option to Work on the App Offline. Wants a functioning free version of the app. </div>		
Demographic <p><input checked="" type="radio"/> Male 25 years</p> <p><input checked="" type="radio"/> Wales, UK</p> <p>Single</p> <p>Media Critic</p> <p>Middle-income</p>	Background <p>Youtuber.</p> <p>Writes Video essays and reviews.</p> <p>He Plays Games and Reads Books in his free time.</p>	
Goals <ul style="list-style-type: none"> • Finding well-written stories to Review. • Growing his Fanbase. • Writing a Novel. 		
Skills <p>Team Management</p>  <p>Social Media</p>  <p>Critique</p>  <p>Gaming</p>  <p>Story-Telling</p> 	Quote <p>“Having something to say is the easy part. But molding and delivering that message, carving it in continuity and pushing every piece alongside it while remembering where and why every element coexists, is where the memory, effort, time and struggle come in.</p> <p>It's where the talent comes in. It's what separates the great.”</p>	
Favourite Stories 		
Channels 	Technology 	

Application Requirements Specification

Asset Creation

A very important part of the story is its “elements” (like characters, MacGuffins, objects of interest, factions, creatures, technology, etc.)

They are needed in world-building and almost every part of the story contains these elements.

So it is essential that the writer can create them and have access to them so they can be used as assets in story writing.

Map Creation

Many stories have complex world-building aspects. From high fantasy with completely original worlds to sci-fi and their faction struggles, world-building is a pillar of storytelling and one of the best ways to introduce readers to a new world is a “World Map”. writers should be able to create a map, containing landscape and natural terrain...alongside roads, borders, and faction territories. They should also be able to place assets on this world map so their location is dynamically visible throughout the plot.

Character Relations

The way characters interact with each other is informed mostly by their History. So keeping that in mind will help with making consistent and solid characterisation. And in many stories where there are tens if not hundreds of fleshed-out characters with a rich history, it is important to keep a neat and visual way of navigating through them. So our app introduces a graphical Web of character interactions with nodes being the characters and edges between them their relation.

TimeLine

Most of the novels on sci-fi and fantasy have an extensive and detailed history to flesh out their world, and in many stories that span over several



books, there is a complex plot that contains many events. So keeping them in mind at all times can be difficult. Our app provides a way so the writer can sort events in a timeline Keeping track of setups and payoffs, plot events, character development, etc.

Script Writing

Writers use plenty of apps to write the script of their stories (word, google docs, LibreOffice, etc).

But our app introduces an alternative where you can link assets to a scene, characters present in the scene, for example, the setups that are written and a link to their payoffs.

Online Interaction

Readers expect to be able to search for books, authors, genres, etc.

When they find a public story they can download the “script” or access the whole project, can see assets, timeline, map, etc.

They also expect to be able to comment on stories or projects, read reviews on it, and also follow and support the writers they like.

Security

Account protection is an essential part of any app with login requirements.

But we also have to consider plagiarism rights. A writer can upload their work and should have ease of mind that their project won't get reuploaded as someone else's work without giving credit. So in the case of story monetization, a basic plagiarism detector should be implemented. From open-source software like “Dupli” and “Copyleaks” to more sophisticated programs like Youtube Content ID, all give us options to use or create a custom plagiarism detection software for the safety of our user's work.

Quality Function Deployment

We can categorise our features and requirements based on user expectations. Features they want present in the app for it to be functional, features that are normal to be in the app and won't gather much praise for it being there, and exciting features, which catch the interest of new users and can potentially help us advertise our app.

After that, we can consider the cost-effective aspects of implementing these features and create a table for their priority.

Expected Requirements:

A secure account login system for accessing their projects (password and 2FA, ways to recover lost passwords, etc).

Having the option of saving the story on a database both local and online (The whole application is meaningless without being able to access previous works).

Creating characters and saving them as an object (characters are the main building blocks of any story so it's important to have an easy yet useful interaction with them).

Taking notes in an advanced fashion (noting makes it clear for the writers to find out their previous views about different parts of the story and not only its existence is important but it should also help them get exactly what they want from it).

A simple and neat graphical interface (a good graphical interface inspires writers and makes them less likely to get bored, and also writing takes a lot of time so having a suitable graphical interface helps their productivity greatly).

Having an online management panel for an Overview of their storylines (one of the main ideas of this app is to help writers have control over what they're writing and to have a clearer perspective about detailed things in their story to not get confused and perceive their work easily).



Users can read a small portion of selling books (users get a sense of what exactly they are buying).

Normal Requirements:

Being able to write the story in an online format (It's important for the interaction between different people and gives users an up-to-date experience and functionality).

Give users the freedom to rank writers and writing scores (it's important to know the other one's previous opinion about something and makes it easier to make better choices for users).

Having a filter for specific genres, writers, stories and writing styles by users.

Having the ability to follow desired writers by users (users of any interactive app need to be able to know more about what they liked previously and following gives them that).

Being able to donate and give financial support to writers.

Having an environment to advertise and sell fan-based products of popular stories (monetization is important to many writers and it helps them greatly to support themselves).

Exciting Requirements:

Drawing World Maps. (world maps give useful information about the setting which is very important in a story).

Saving the relationship between different characters and visualising it in a comprehensible graph.

A timeline feature so the writer can sort events in a visual timeline to help them keep track of plot events, setups and payoffs, character development, etc.

Defining asset templates that can be easily used in future (it saves a lot of time and also makes it possible to have better assets as time goes by).

Predefined templates for different writing styles (different writing styles also have different features in their appearance so having different templates is useful, some provided by us, others by the community).

Merging the stories (sometimes writers find it useful to use the element of another story on their own).

Users and writers have the option of interacting with each other like a social platform.

Creating groups of writers and the ability to request and join them (sometimes different people want to write a story together and it's important to give them that ability).

	The Feature	Priority
1	A secure account login system for accessing their projects.	High
2	Project saving both local and in a database.	High
3	Creating characters and saving them as an object.	High
4	Having an online management panel for an Overview of their storylines.	High
5	Being able to write the story in an online format	Medium
6	Give users the freedom to rank writers and writing scores	Medium



7	A basic social platform for users, ability to follow and rate user posts.	Medium
8	Drawing World Maps.	High
9	Character relation graph.	High
10	Story TimeLine.	High
11	Predefined asset templates for different writing styles.	Low
12	Merging the stories.	Low
13	Being able to donate to support the writers	Low
14	Creating groups of writers for shared projects	Low
15	Opportunity for the writers to advertise and sell fan-based products of their stories	Low

Channels and Budget

The initial fund is contributed by sponsors and public support. The growth of the app is obtained by Publishing commissions, story purchases, donations, and in-app advertisements.

We advertise our app using social media advertisements, sponsorships to relevant content creators, and Youtube Channel influence.

Links:

Opportunity Board:

https://miro.com/app/board/uXjVO5TmXE8=/?share_link_id=761550154982



Use cases

We create use case diagrams for each main part of a project (Creation and Deployment). And represent scenarios in use case diagrams:

- 1: Process of writing a story from the writer's point of view.
- 2: An overview of the Application for all users in the deployment process. (Connection of publishers with stories and writers.
A reader who finds and reads stories and follows writers.)

Use case 1: Creation Process

Use case: Project Creation

Primary actors: Writer/Creator, Artist

Goal: How the Writer Creates the story.

Preconditions: The Primary actor has logged in to this account at least once in the recent past.

Trigger: New project is created.

Scenario:

The writer started the project.

The writer creates assets and story elements.

The writer creates a world map.

The writer creates a plot timeline.

The writer determines character relations.

The writer writes story scenes.

Exceptions:

The actor has not backed up their data and the local files are deleted.

The actor has automatically logged out after two weeks of inactivity.

Priority: Essential.

Available in the first increment.

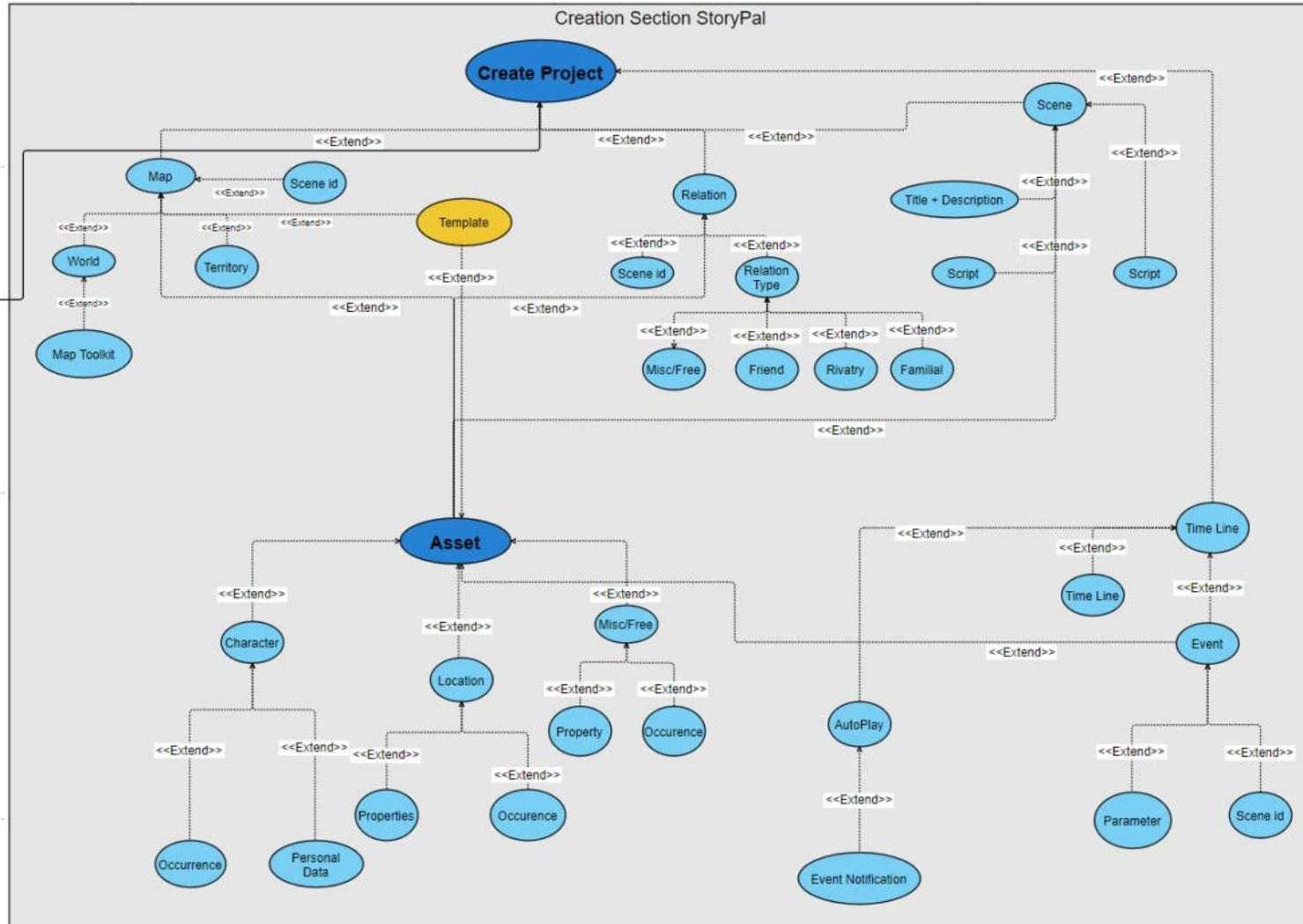
Secondary actors: none.

Open Questions:

Should we save the world map as a picture or specialised format?

How should the character relation graph be represented?

Which features should be available offline?





Use case 2: Deployment Process

Use case: Project Deployment Overview

Primary Actors: Writer/Creator

Goal: how the actors interact with a created story.

Preconditions: all actors are signed in and our servers are functioning properly.

Trigger: A story is completed and uploaded.

Scenario:

Primary actors are logged in.

The project is created by the writer.

The project is uploaded by the writer.

The project is verified and accessible by other users.

Exceptions:

Our Database is down. The actor isn't logged in.

Priority: High.

Available in the first increment or the first major patch.

Secondary Actors: Publishers/Editors, Readers

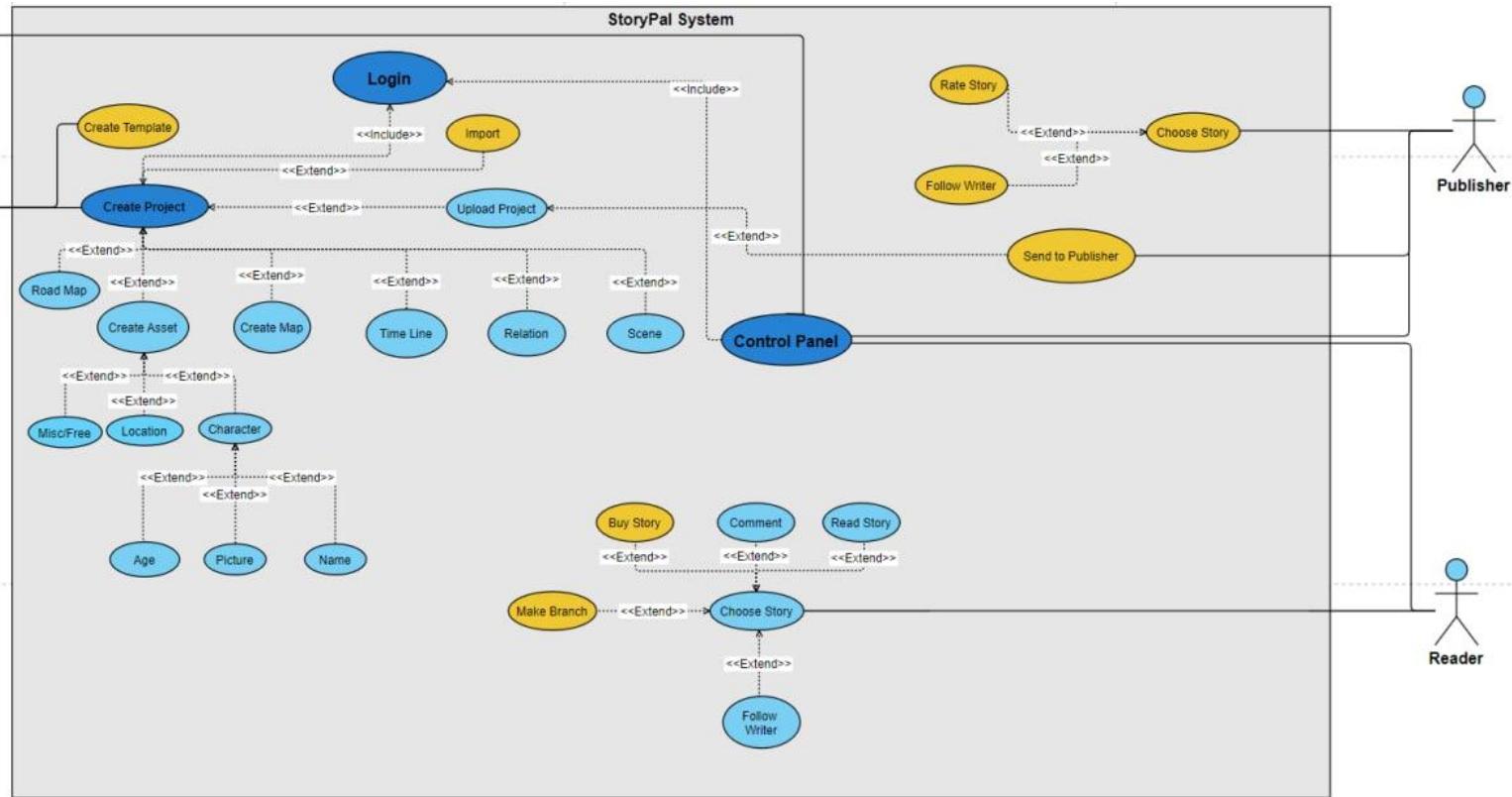
Open Questions:

How much of the project do the readers have access to?

How can the writer monetize the project?

Should we implement a way for a publisher to buy the right to publish a story?

Should the writers be able to team up?



Use case 3: Reader Experience

Use case: Reader Experience

Primary Actors: Reader

Goal: How the reader navigates and uses the app.

Preconditions: All actors are signed in and our servers are functioning properly.

Trigger: A reader decides to use the app.

Scenario:

Primary actor logs in.

They search for a story.

They read a portion and decide to purchase the story.

They leave a review.

Exceptions:

Our Database is down. The actor isn't logged in.

Priority: High.

Available in the first increment or the first major patch.

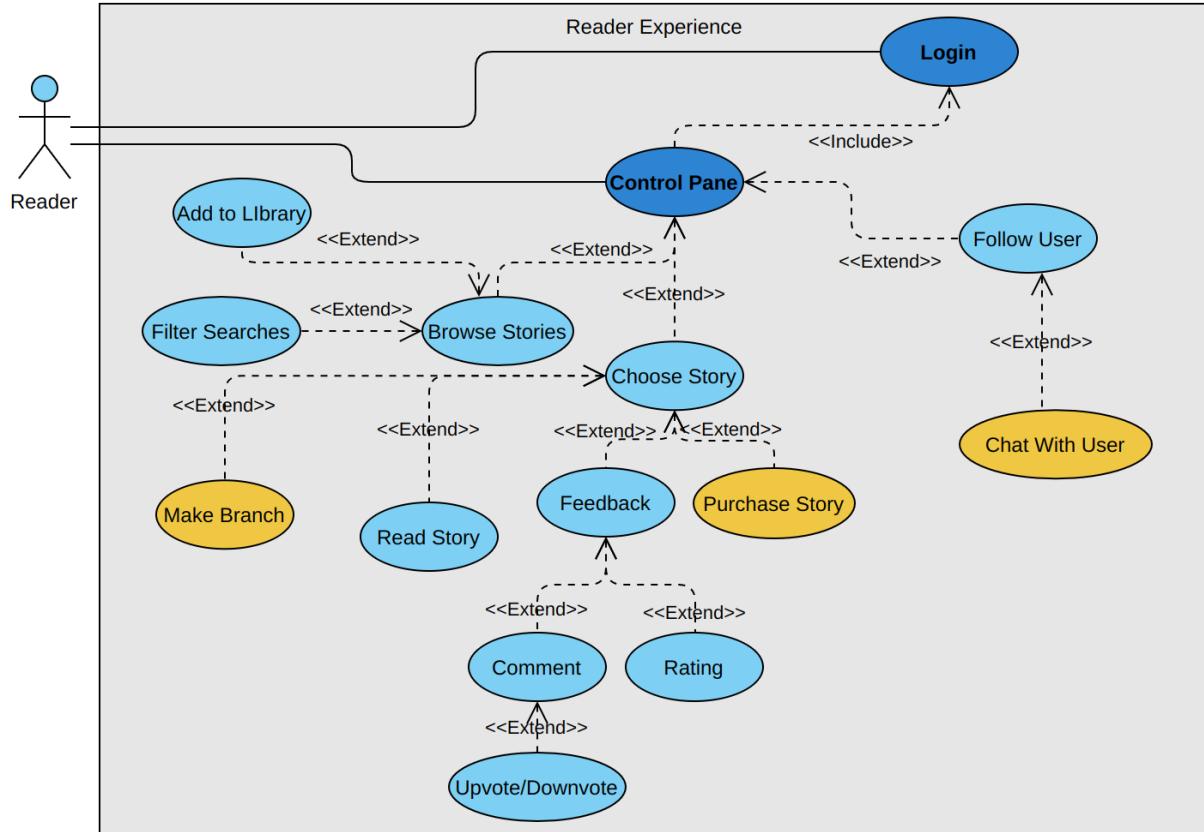
Secondary Actors: none

Open Questions:

How much of the project do the readers have access to?

How in-depth should the user interaction (social platform) be?

Should we implement a story purchase system in our app for readers?





Use case 4: Publisher Experience

Use case: Publisher Experience

Primary Actors: Publisher/Editor

Goal: How the publisher navigates and uses the app.

Preconditions: All actors are signed in and our servers are functioning properly.

Trigger: A publisher decides to use the app.

Scenario:

Primary actor logs in.

They search for a story.

They read the story and leave a review.

They find a story they want to publish.

They contact the writer for an agreement.

Exceptions:

Our Database is down. The actor isn't logged in.

Priority: Normal.

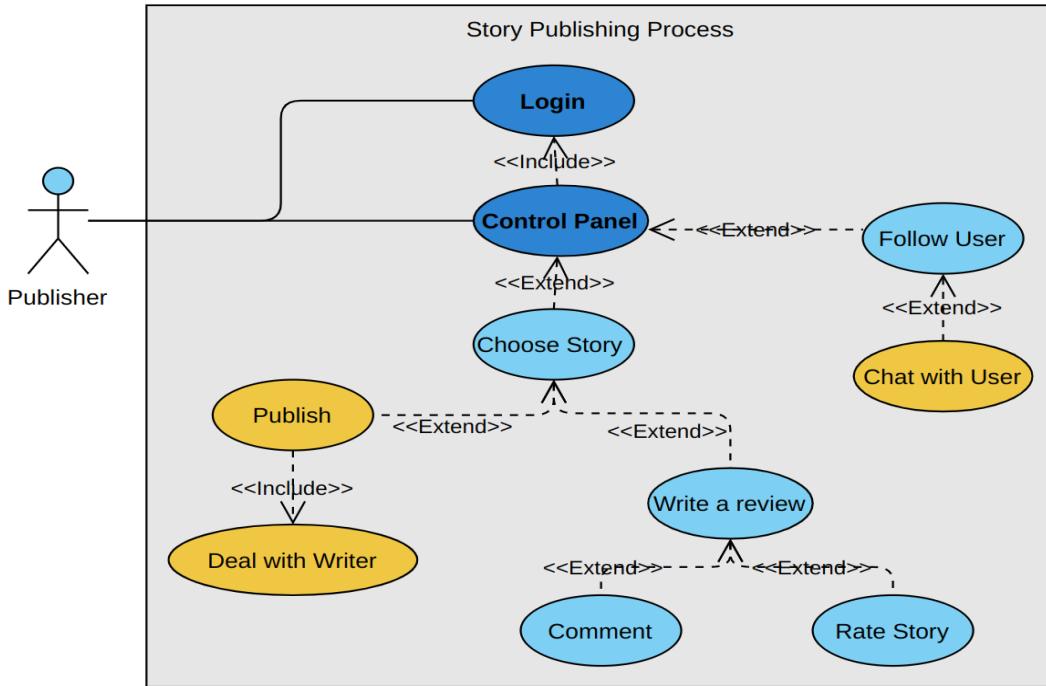
Available in the first major patch.

Secondary Actors: none

Open Questions:

Should the contract between writer and publisher be conducted in the app?

How can we moderate these interactions to prevent scams?



Legend:

DARK BLUE High Priority, Necessary for accessing the app features.

CYAN Normal priority or are a subfeature of another feature.

YELLOW Low priority or are not cost-effective to implement in the first iterations.

Software Requirements Specification (SRS)

In previous sections, we established features that are required for our application along with stakeholder expectations on how the app should function.

Now we examine the software requirements and plan out a strategy for the implementation stage.

Our goal is for a desktop version of the application to be released for Windows, Macintosh and Linux systems at first, and a web version to be developed in later renditions.

Front end

Our application focuses on user interaction with the software and a clean graphical interface. So Front-end is a high priority.

Based on software needs and the developing team's experiences we will use the C# language For the desktop version of the application.

We believe C# is a powerful language suitable for our needs and its ease of use will speed up the development process. And since it is backed by Microsoft, if the need for additional assets in the developing process arises, we can find help for our needs.

For the web version of our application, the front end will be developed with HTML, CSS, Sass, bootstrap and javascript using the React Framework as they are the standards and our team is experienced in these languages.

Back end

For the backend of our application, we encounter certain challenges.

Thankfully our app does not require much server usage in the first place so all we need is a functioning external site for importing and exporting light files.



Furthermore, in the first implementation, the app doesn't intend to support transactions and purchases but after a couple of patches, we have to look into integrating the app with a direct banking interface of a virtual wallet like "AsanPardakht".

Secondly, the "World Map" feature of our app. Which needs a dynamic take on maps. Users should be able to draw landscapes, borders, roads, and territory; They should also be able to update them and place assets (characters, objects of interest, etc) on them based on the story.

And the final feature of interest is the "Character Relations".

Our application needs to help the user easily analyse the relations between the characters in the span of the story. That is why we plan to implement a graphical interface, neatly visualising these relations in a graph, with nodes being the characters of our story and the edges being the relation between them accompanied by links to the reference scenes.

Aside from these two features the backend of our application is fairly simple. In the desktop version, we will use the C# for its ease of use and safety and easy compatibility with the application front end.

For the web version, we have the option to use Python with the Django and Django Rest Framework and PostgreSQL as it suits the application needs and team members have experience with this language which helps the development process.

Figma MVP

At first we present our Figma MVP, aiming to give an understanding of how the app will look like when completed alongside a link to a prototype to simulate the experience. Then we explore those interactions with a state diagram for the front end. And afterwards, delve into the structural diagrams for the project.

The Link to the Figma prototype:

<https://www.figma.com/file/P1NcvRs7lcyl8VabU64YX/Figma-mvp?node-id=0%3A1>

For writing and user interaction with the app we have chosen a dark theme for the primary colour theme and we took inspiration from a colour palette on [colorhunt.co](#) and used it as a baseline for our colour theme.

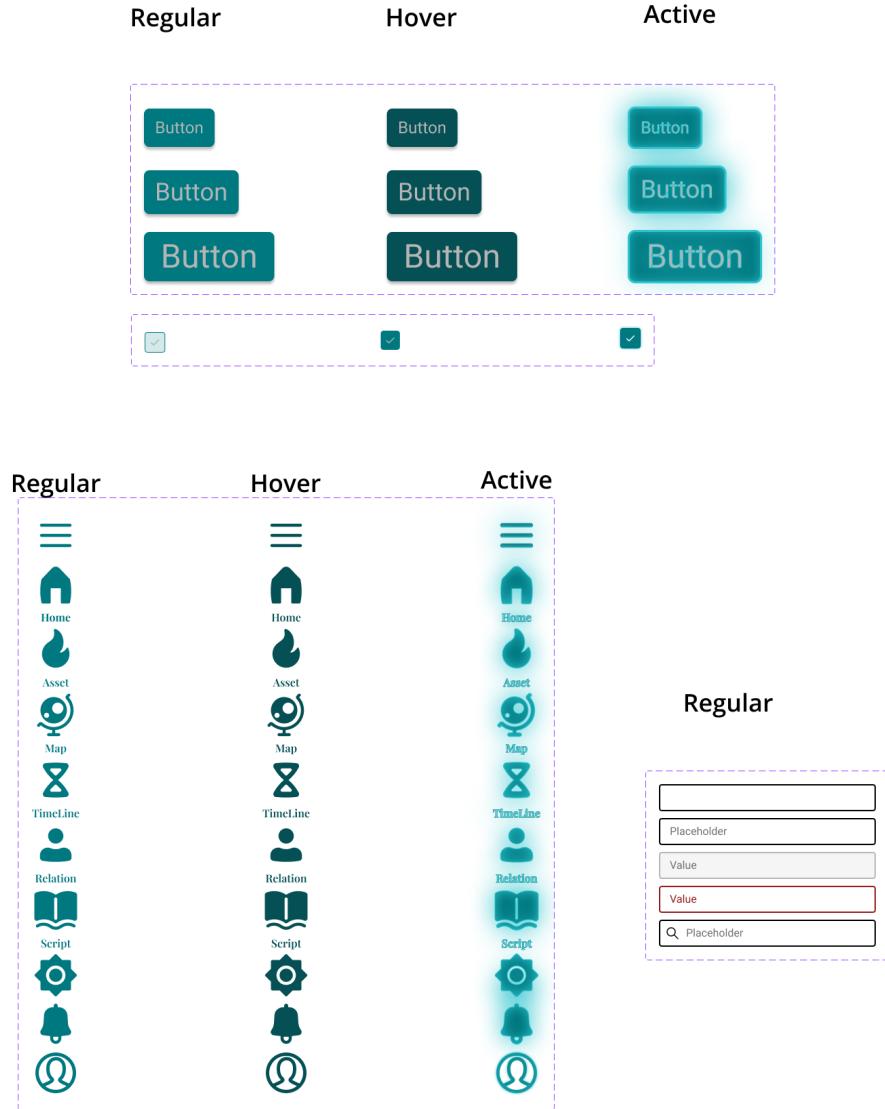
The link for the colour palette is provided below:

<https://colorhunt.co/palette/222831393e4600adb5eeeeee>

Afterwards we cross-referenced the palette with [Colouring for the colourblind](#) for any problems with the visually impaired and found satisfactory results for deutanopia and tritanopia.

Finally using the “A11y - color contrast checker”, we checked and adjusted the foreground and background colours for easy readability.

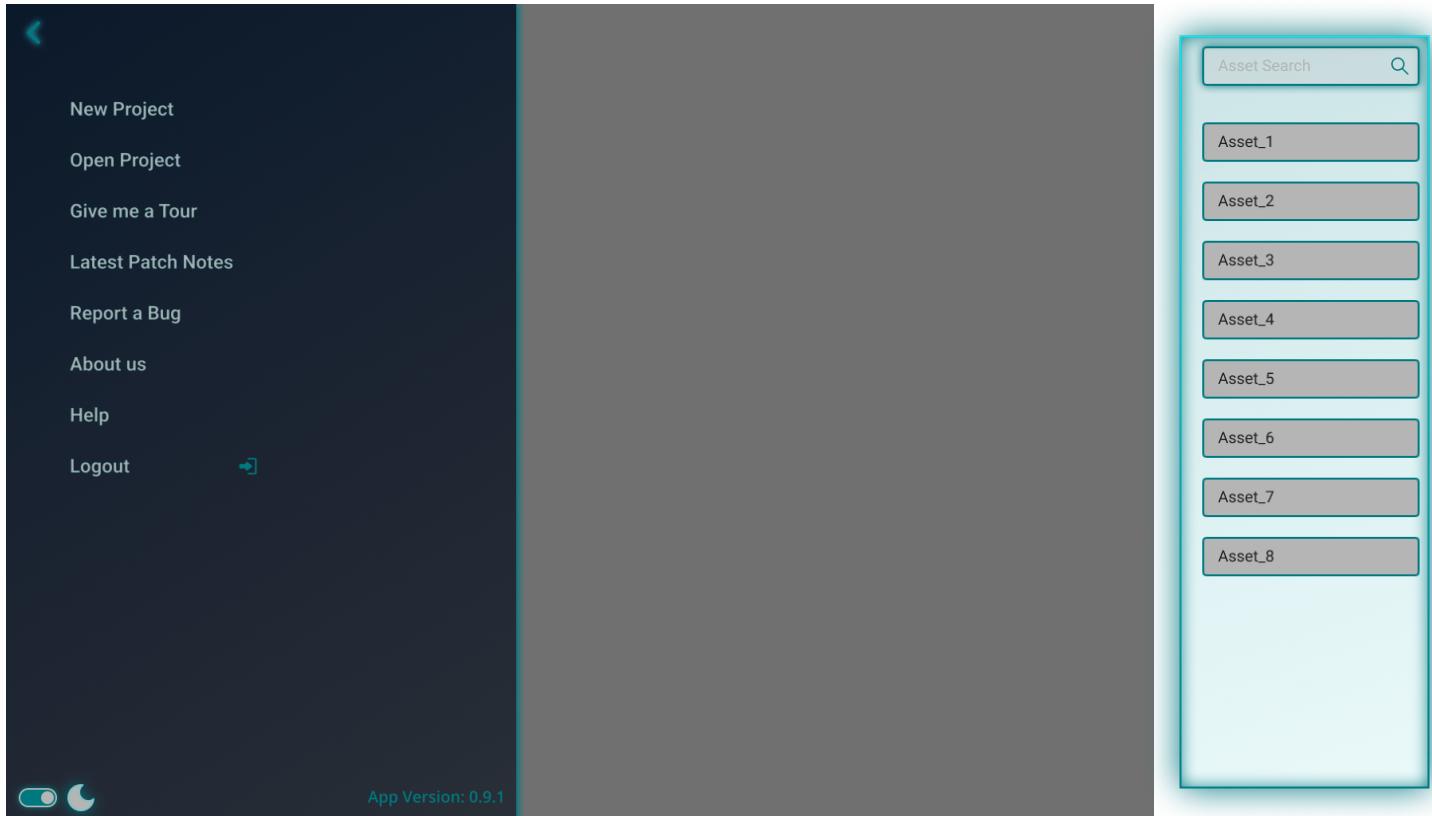
The design focuses on a neat and simple interface to not distract the writer. While using technics like glassmorphism and neon to lighten up the design and give it depth.



Variants

For the efficiency and consistency of the app. We designed components and their variations so it would be possible to use them with a few simple clicks rather than redesigning from the ground up. And it will help with consistency and draws a guide to stick to when designing the app.

The main variations are “default”, “hover” and “active”, which represent the state of the component when it is normally. when it is being hovered over by a mouse, and when it is pressed and activated.



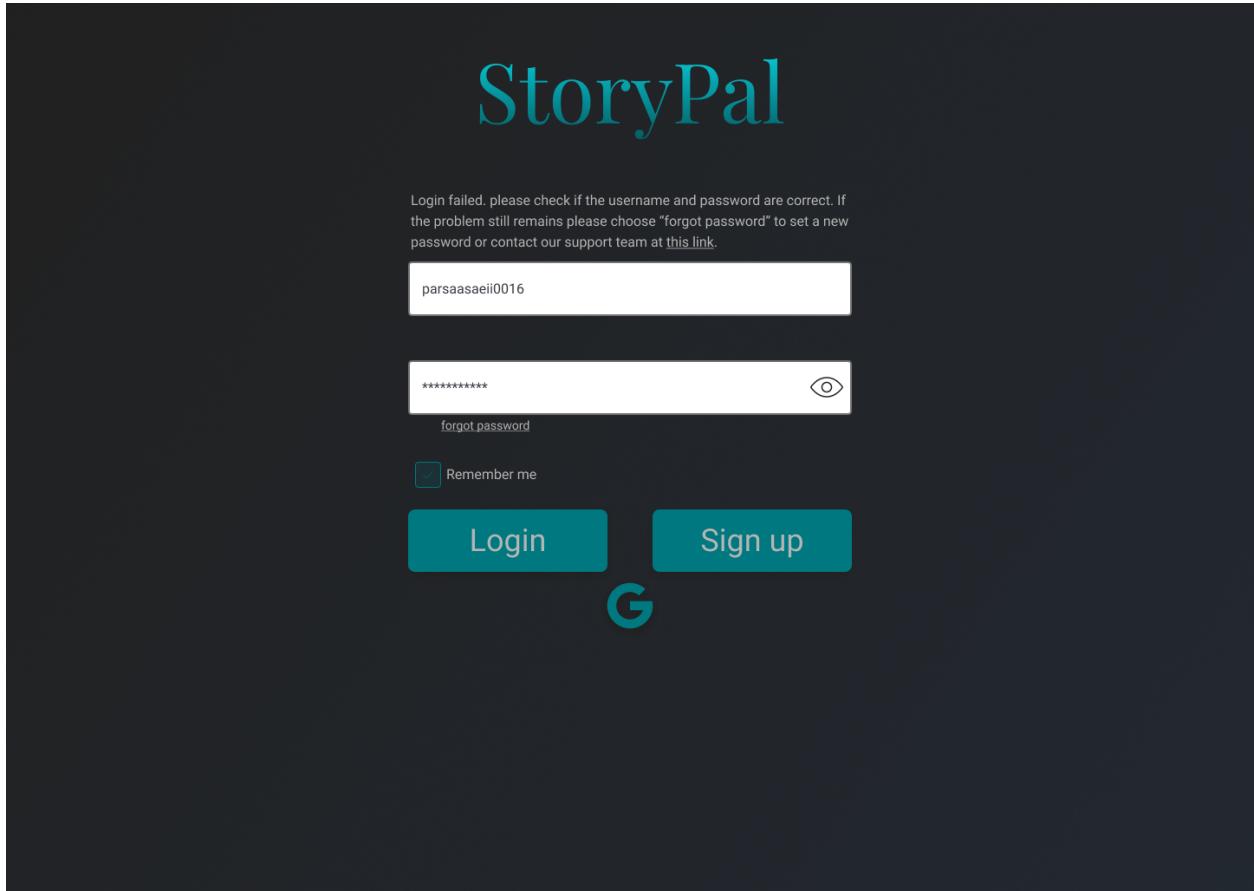
Complex Components

Then we retroactively made some templates for more complex components. Whether they were a pop-up message with a glass effect or a side menu with many options. We decided to make some components and use them when the need arises even if it is needed only once.



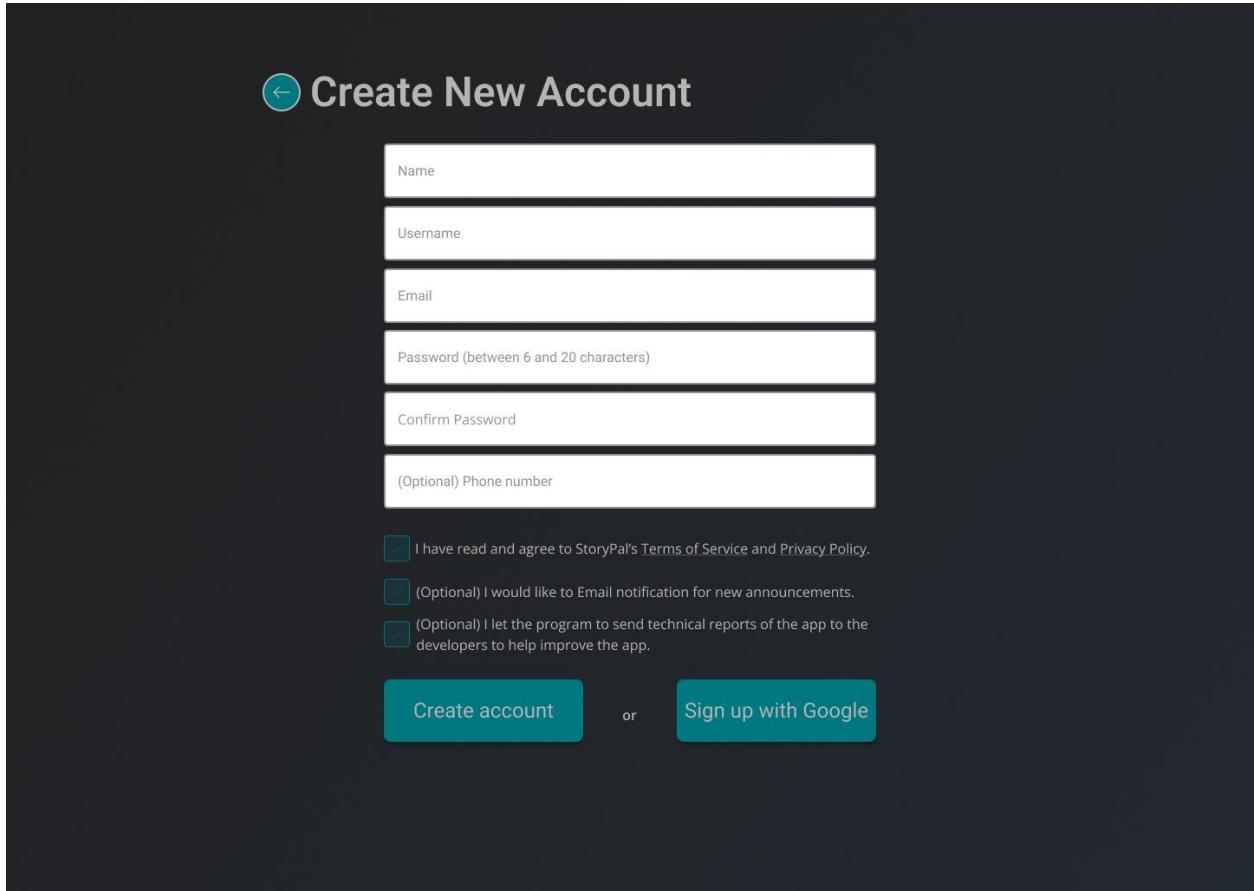
Introductory Page

This is the first page the user will interact with our application. The design is simple and allows the user to directly enter their account. They can either log in or use their google account to easily access their projects. They also have the option to check the “Remember me” button to skip this step for the next sessions, which will be restarted after 2 weeks of inactivity for security purposes. They also can Sign up and get redirected to create a new account, or click the “forgot password” button for quick and easy password recovery.



Login Fail

If the user provides incorrect login info they will be greeted with a fail login message suggesting they try again or use forgot password.

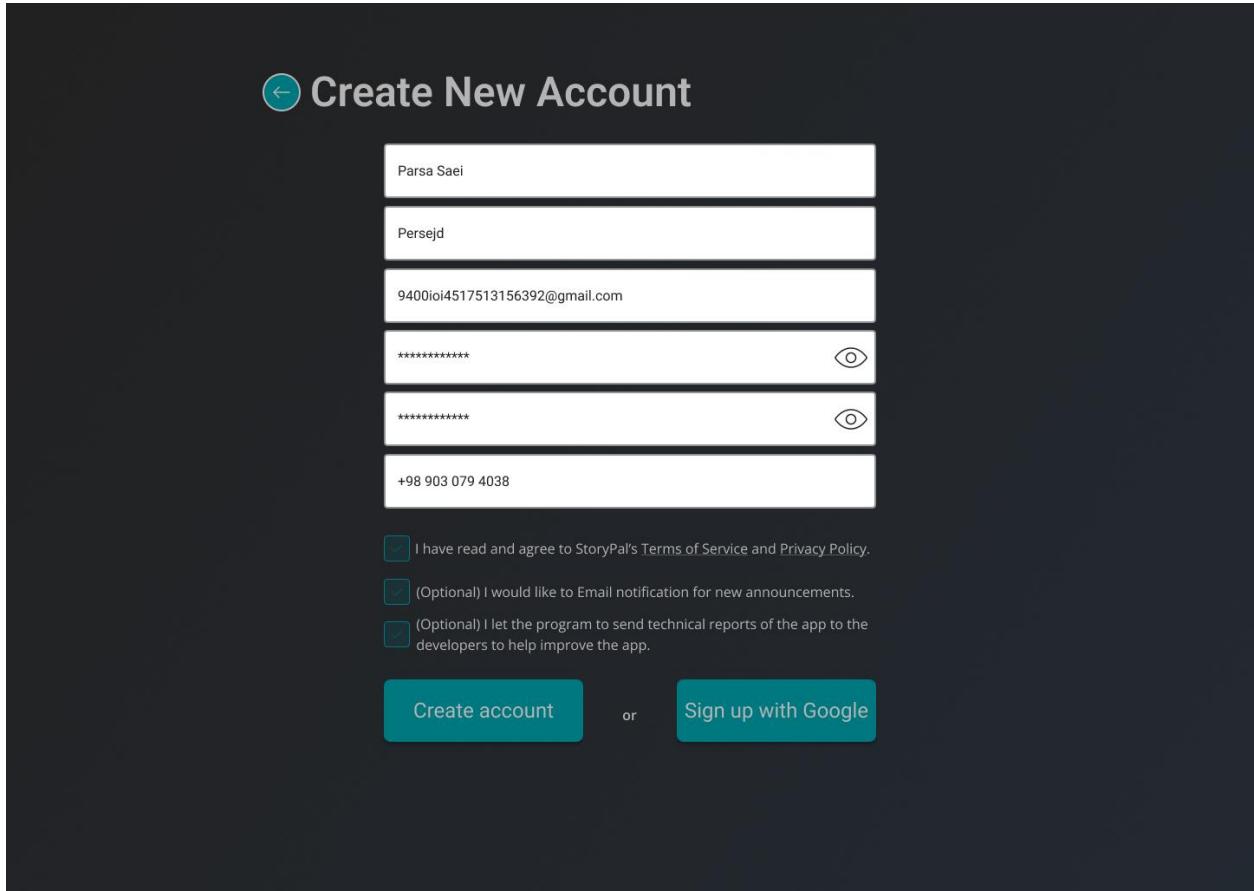


Create New Account

On this page the user can create a new account or use their google account for easy sign-up.

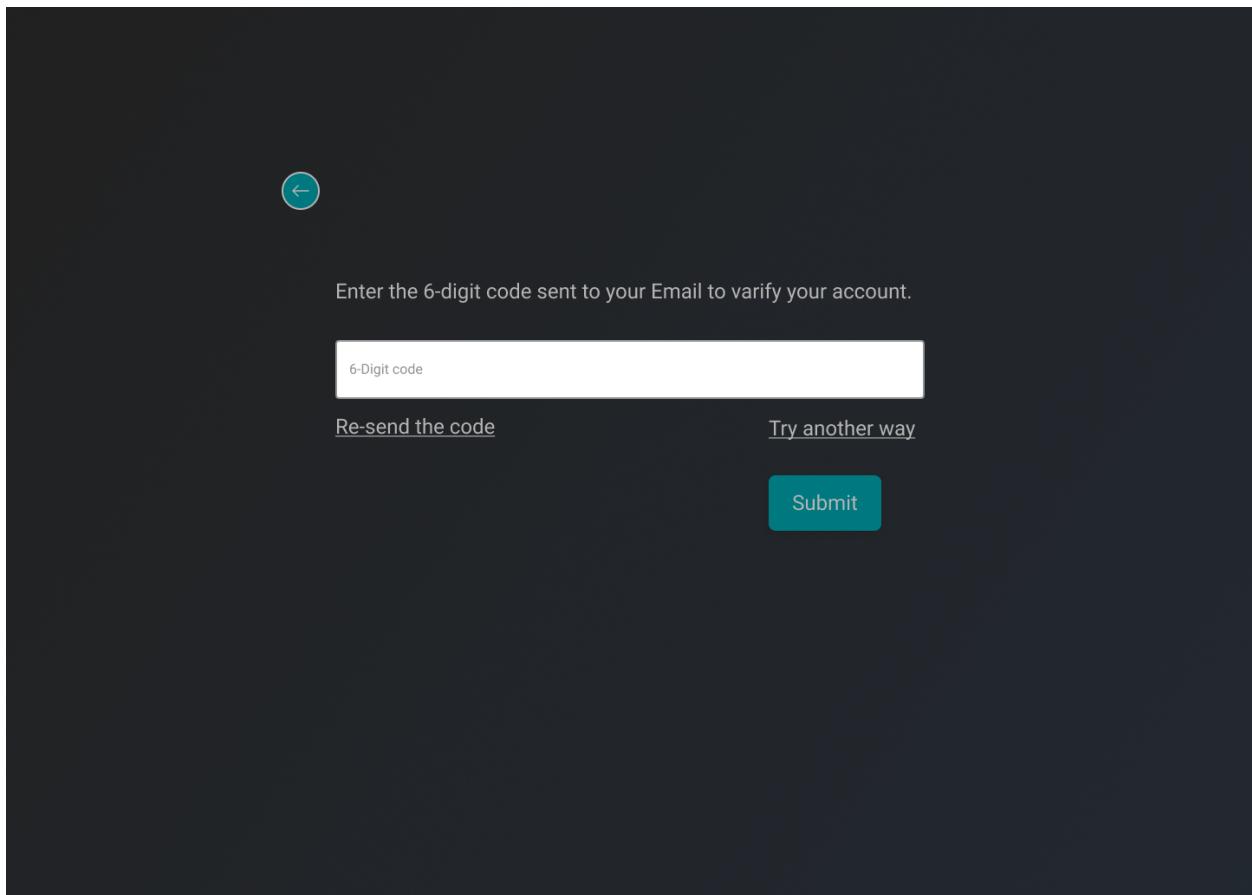
They are required to provide basic security information like username and password and can optionally provide a phone number for easier contact with others and security backup.

After that, they need to read and agree to our ToS and Privacy Policy.



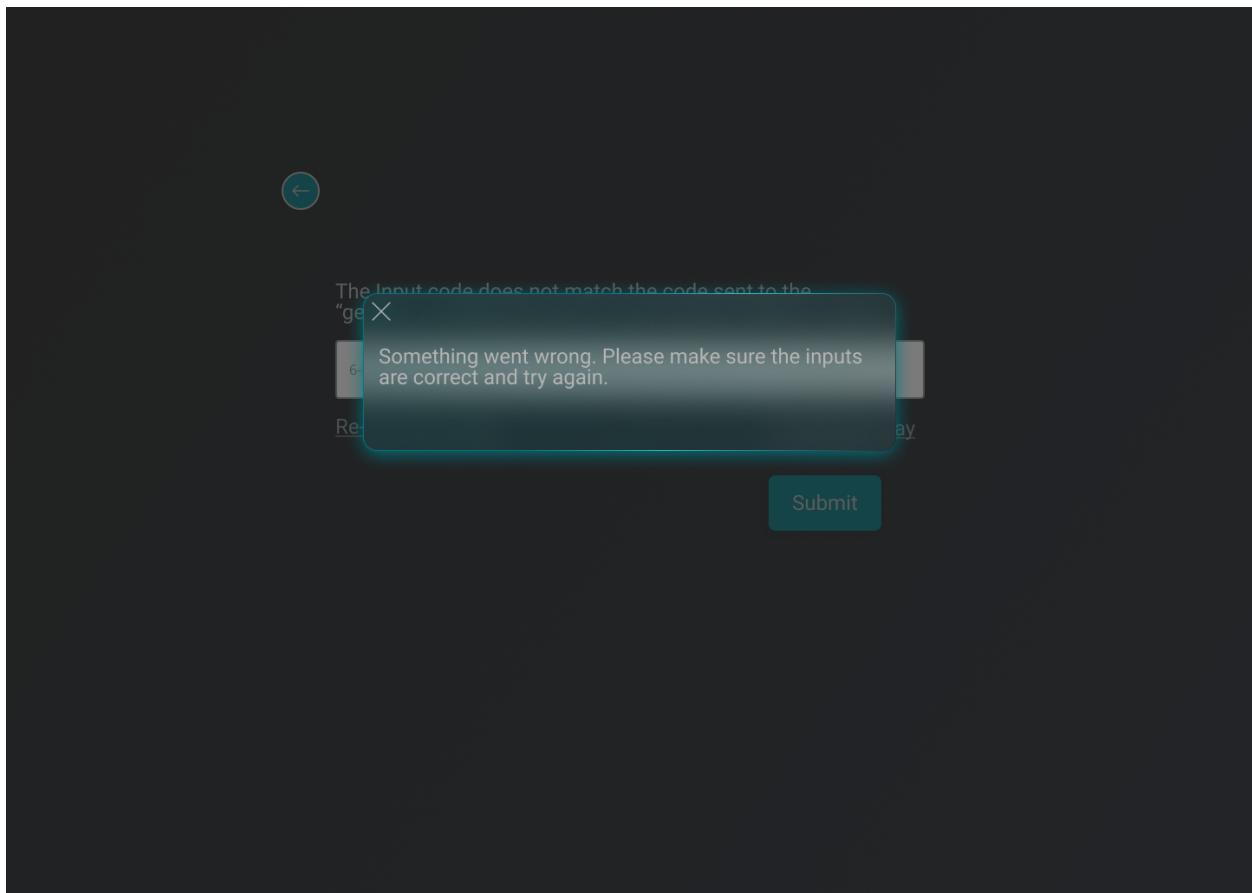
Create New Account Example

Here is an example provided for filling out the page.



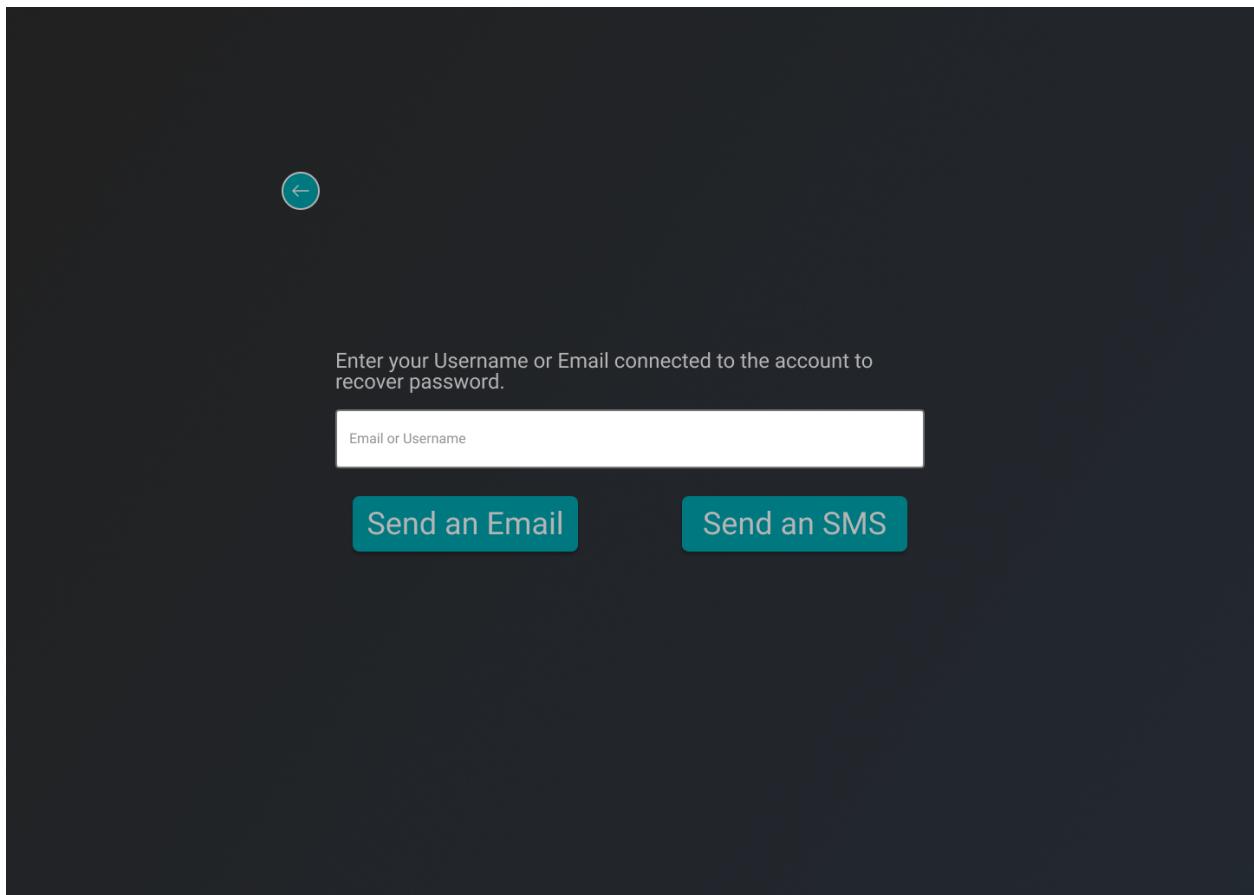
Sign-up Verification

After they provide the needed info the app will send a code to the email stated for verification before creating a new account. This will help against bot accounts.



Verification Failed

If a problem arises the user can request the verification email to be re-sent or they can try another way like a google account.

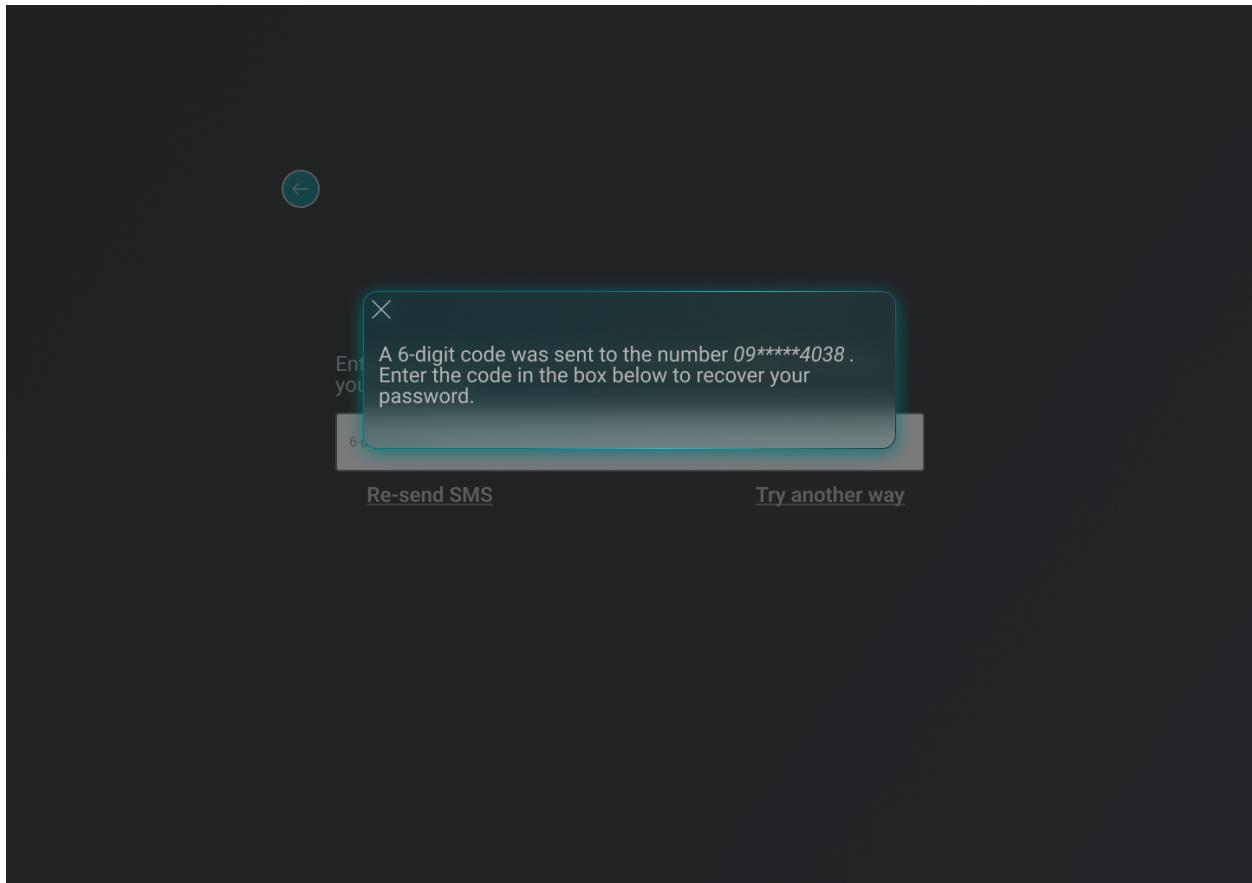
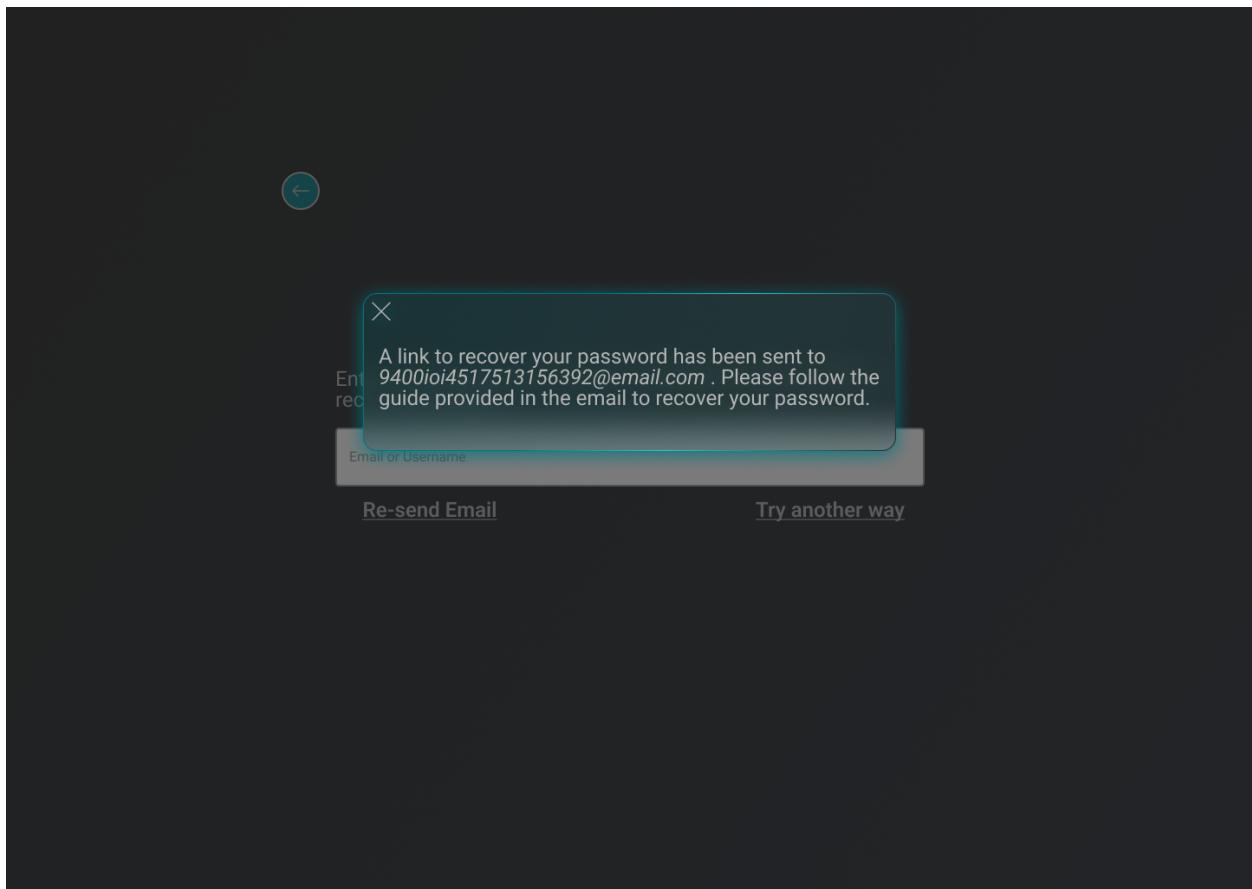


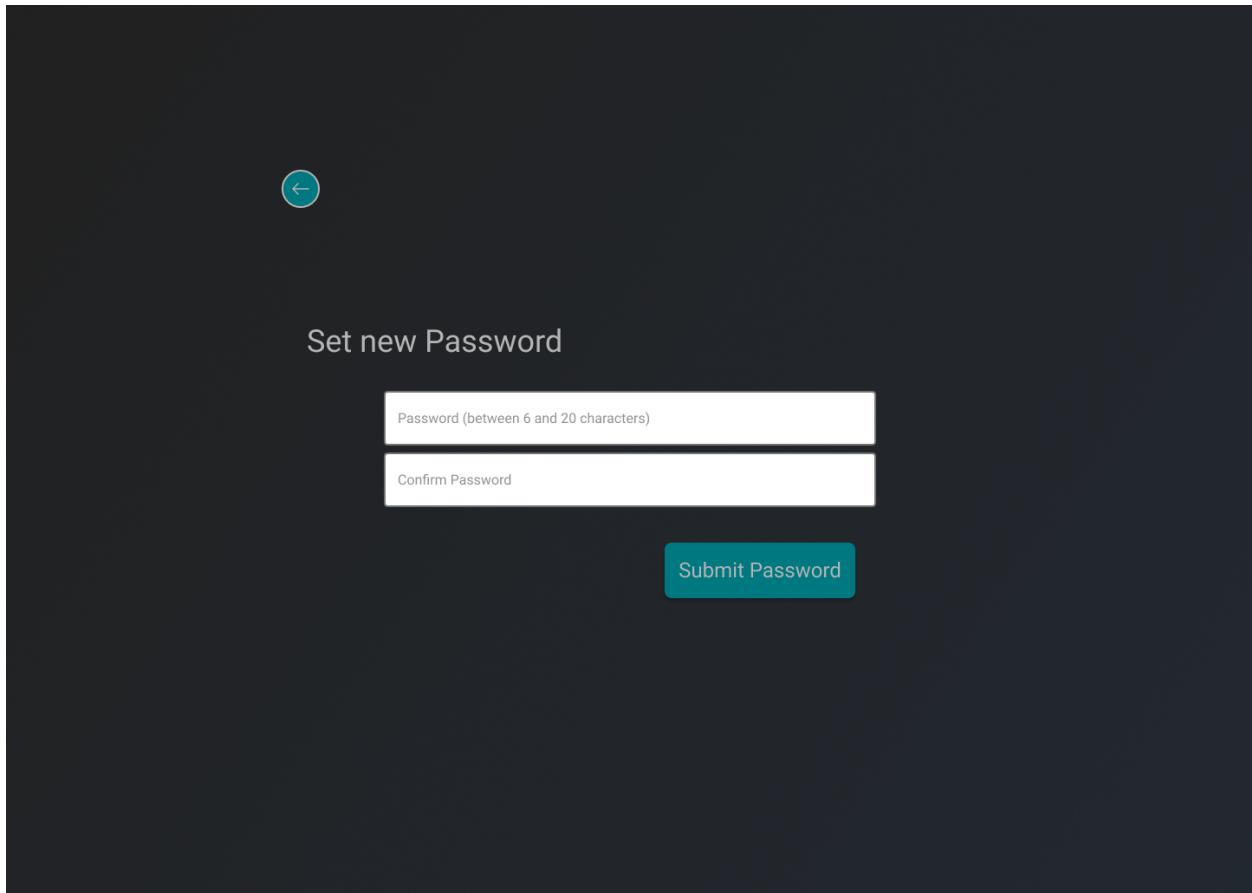
Forgot My Password

If the user forgets their password they can click the forgot password on the initial page and be redirected here to set a new password.

After they enter the account username or account email they can receive a password recovery Email or if they have provided a phone number they can receive an SMS instead.

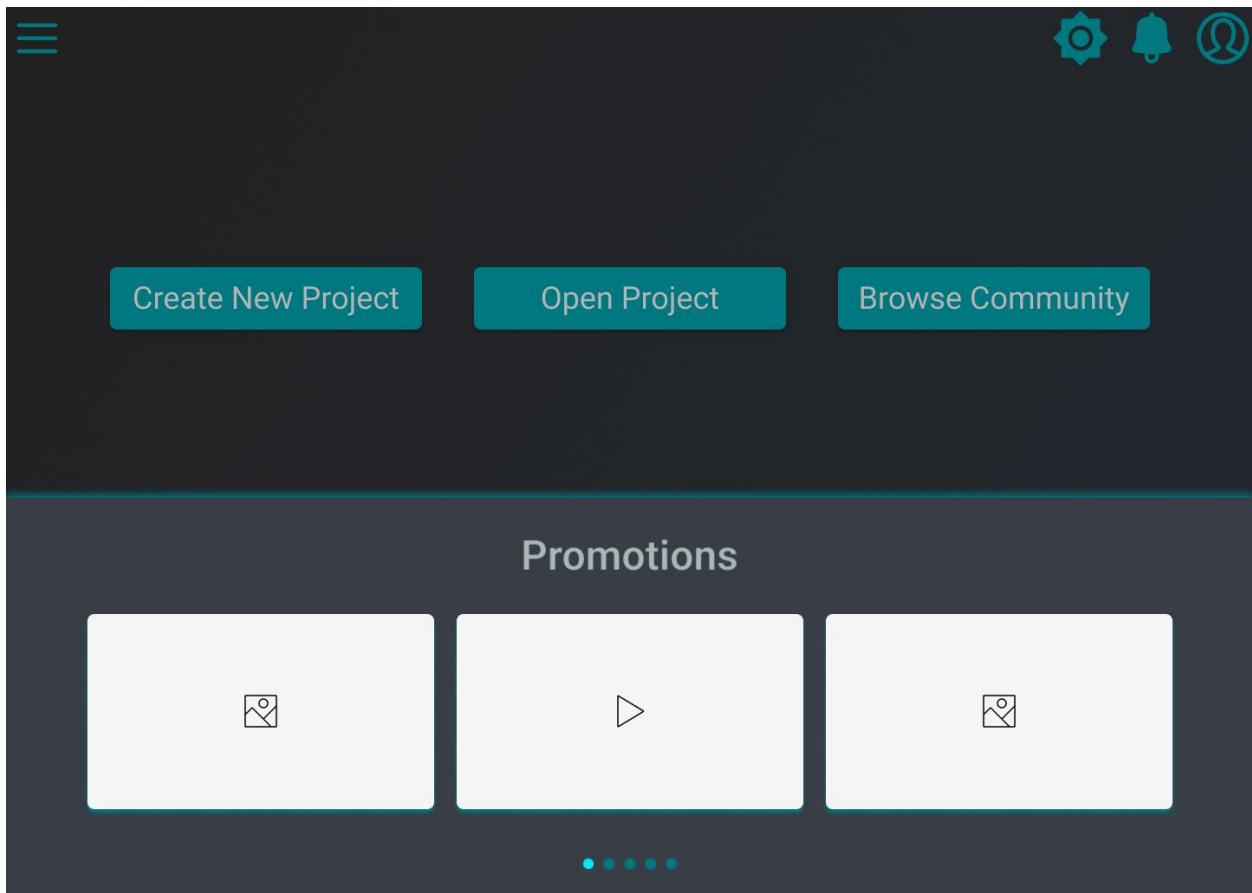
They will receive a confirmation message that the Email or SMS has been sent, with instructions on what to do after.





Set New Password

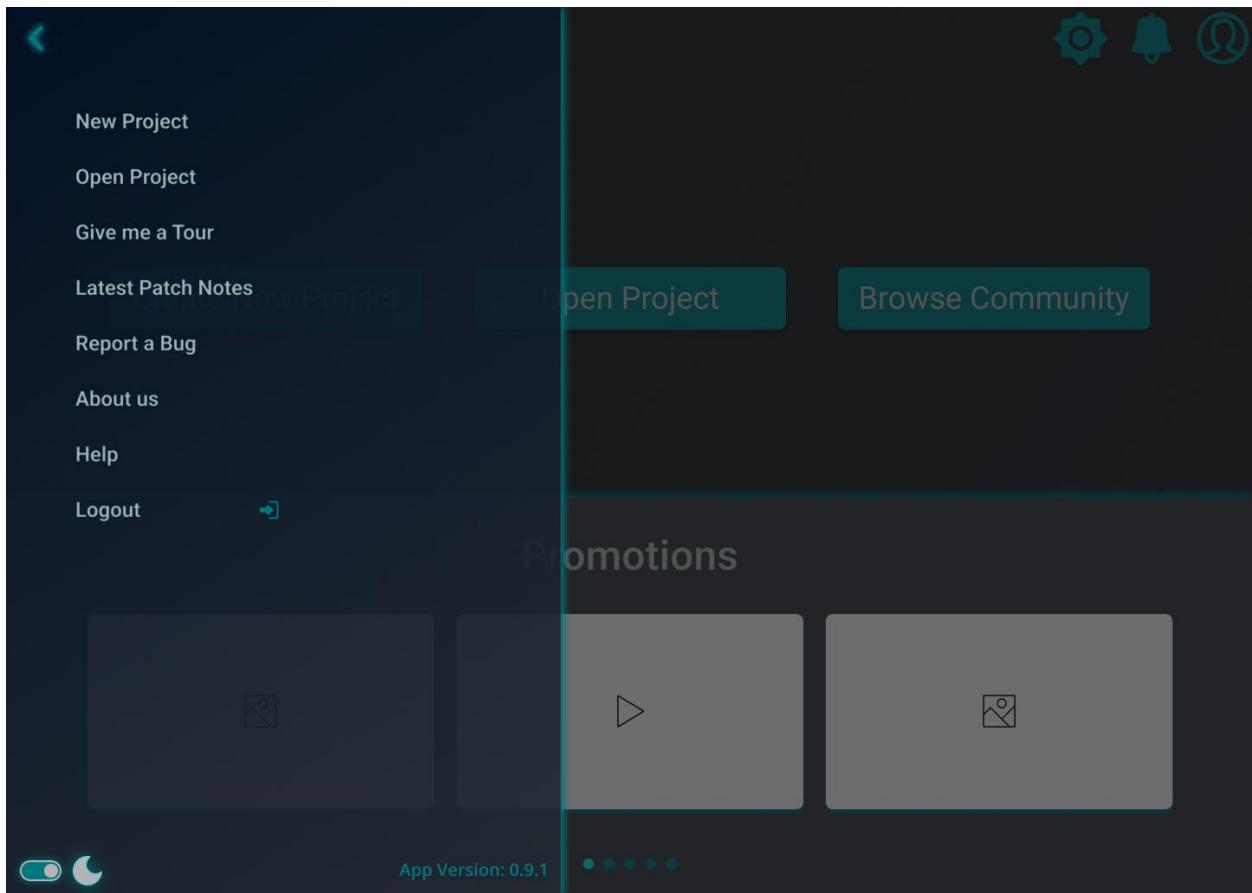
After the user clicks the email link given or enters the 6-digit code provided by an SMS, they can set a new password on this page and get access to their account.



Home Page

The “Home Page” is a hub area where the user can manage their projects, create new projects, browse and interact with a community of creators and readers.

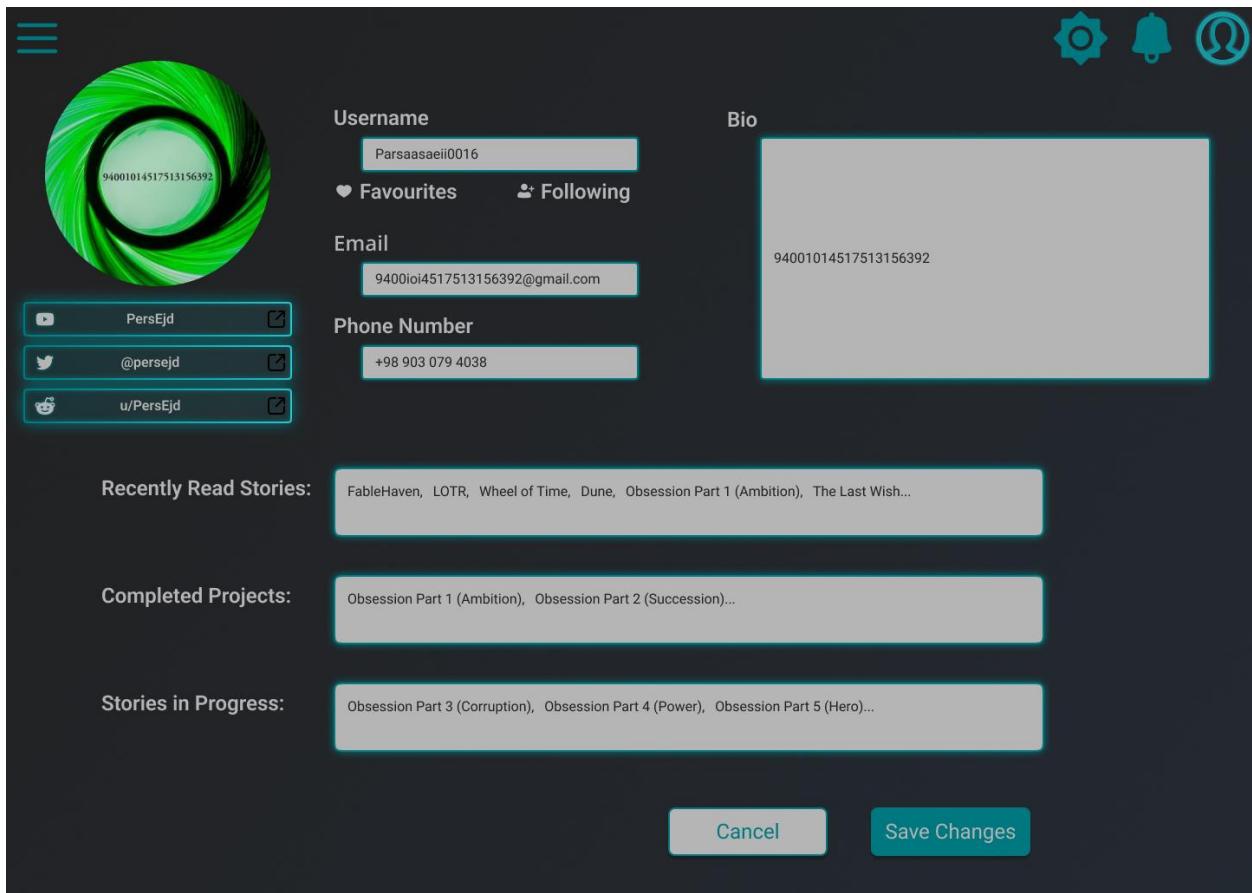
At the bottom, you can see the “Promotions” zone. Where we can recommend well-made projects, our best users or even advertise.



Side-bar Menu

This is a shortcut layout intended to be accessed from any page by pressing the menu button on the top left of the page.

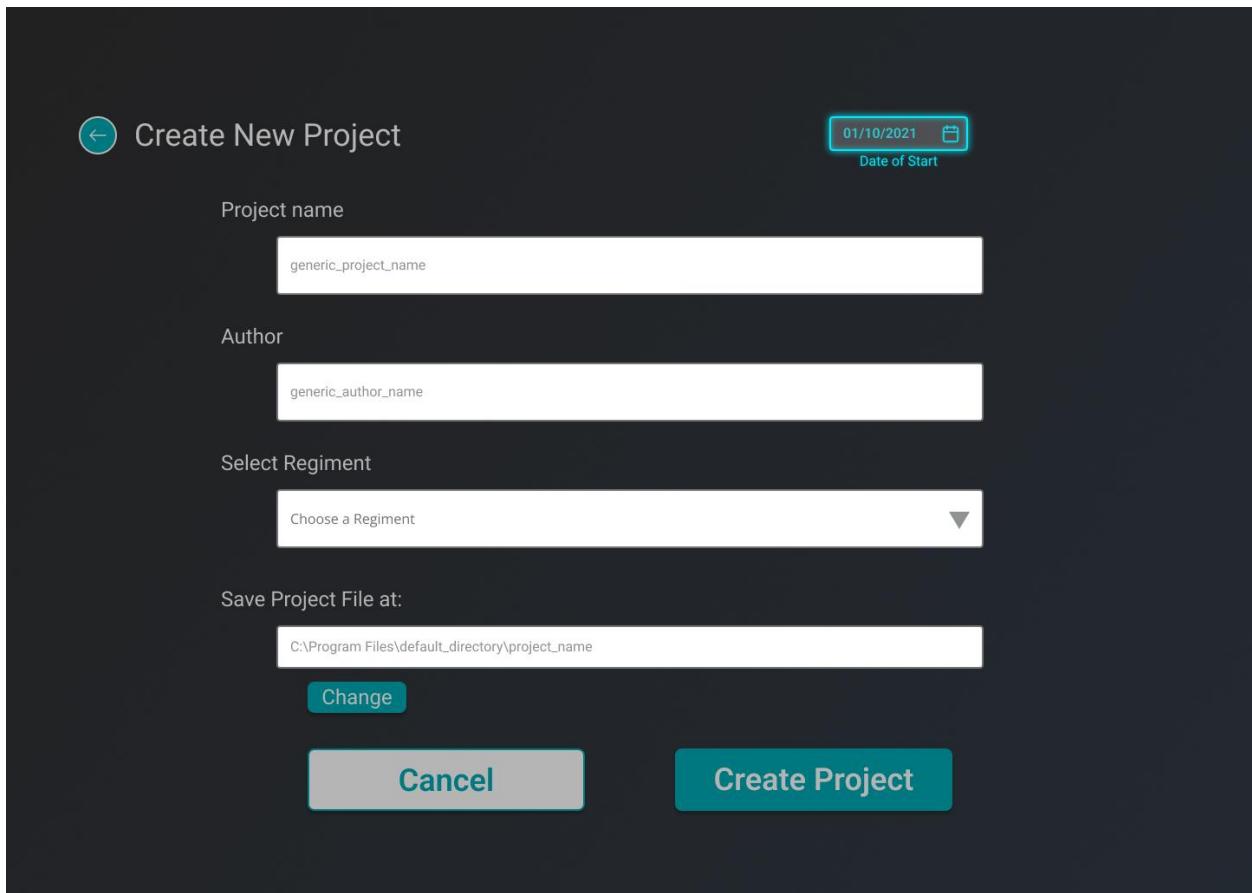
It provides quick access to projects and it can be used to receive info about the app and its creators.



User Profile

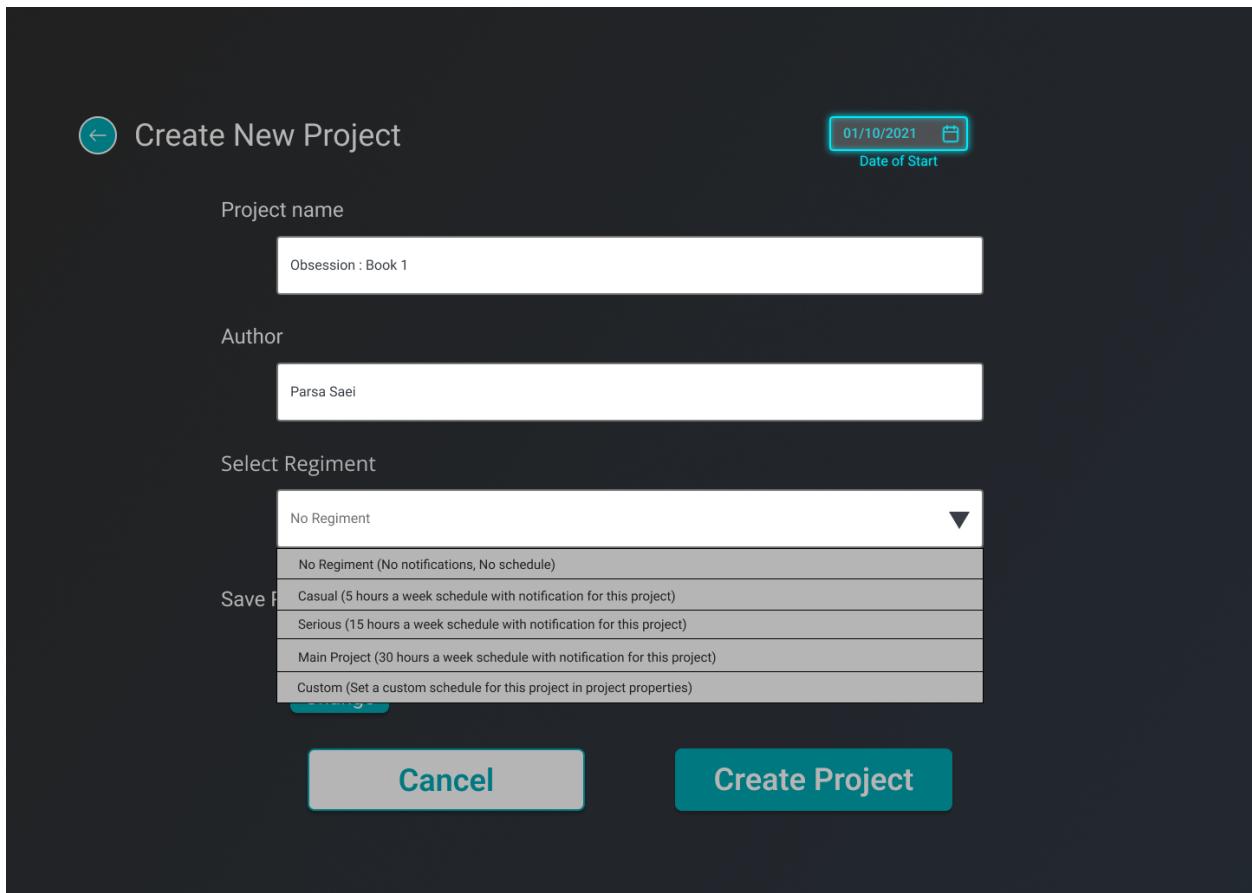
On this Page the user can access and edit their basic information like Bio, username and other fields.

They can also see their activity and project history alongside their channels on other platforms.



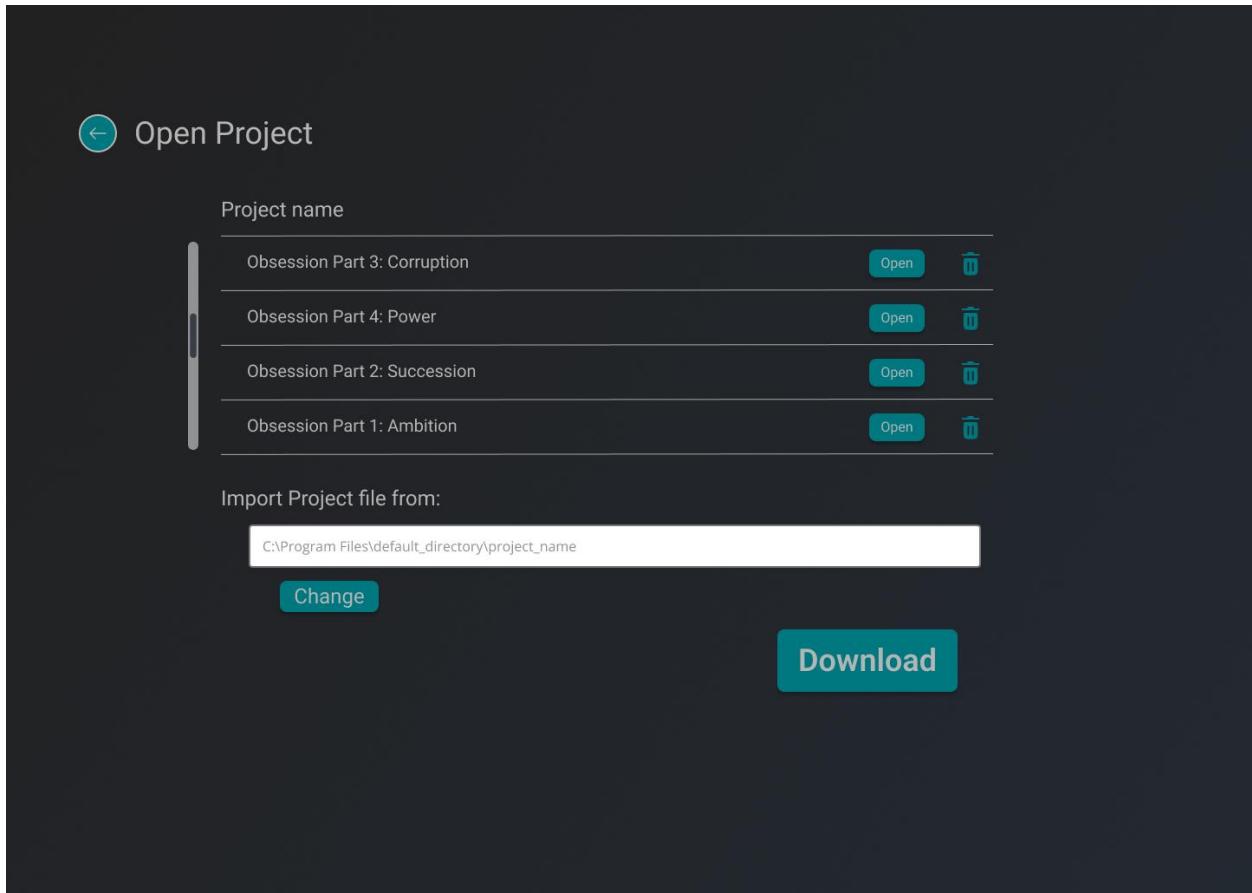
Create New Project

From the home page, the user can access the “Create New Project” page. The button will redirect the user to this page where they can create a new project and choose a regiment for a schedule and notifications and finally save it at the desired destination.



This is an Example provided for a new project.

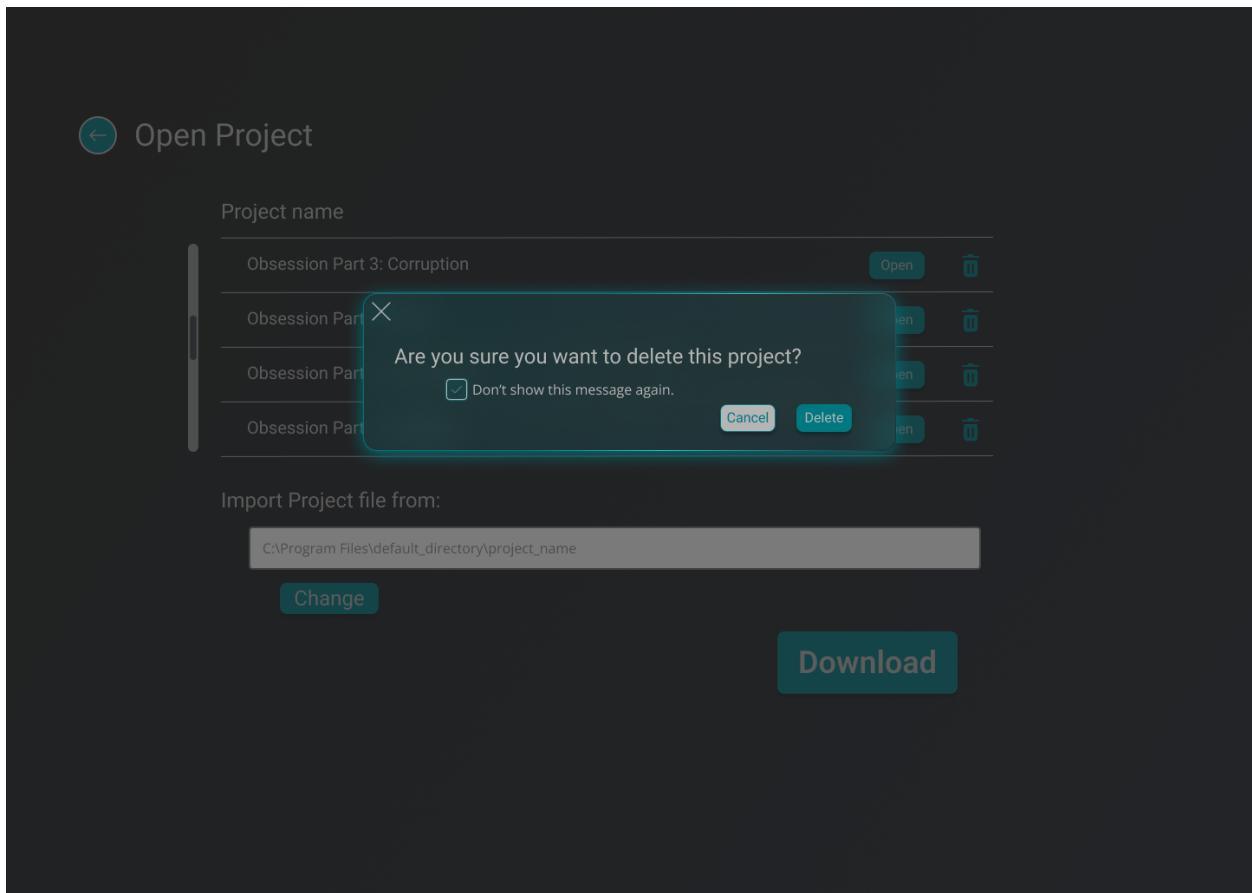
The user can choose a regiment and the app will notify them based on the intensity of the regiment.



Open Project

The user can open and work on a previously made project by choosing their project on this page or importing it from a local file or even downloading a community-made project.

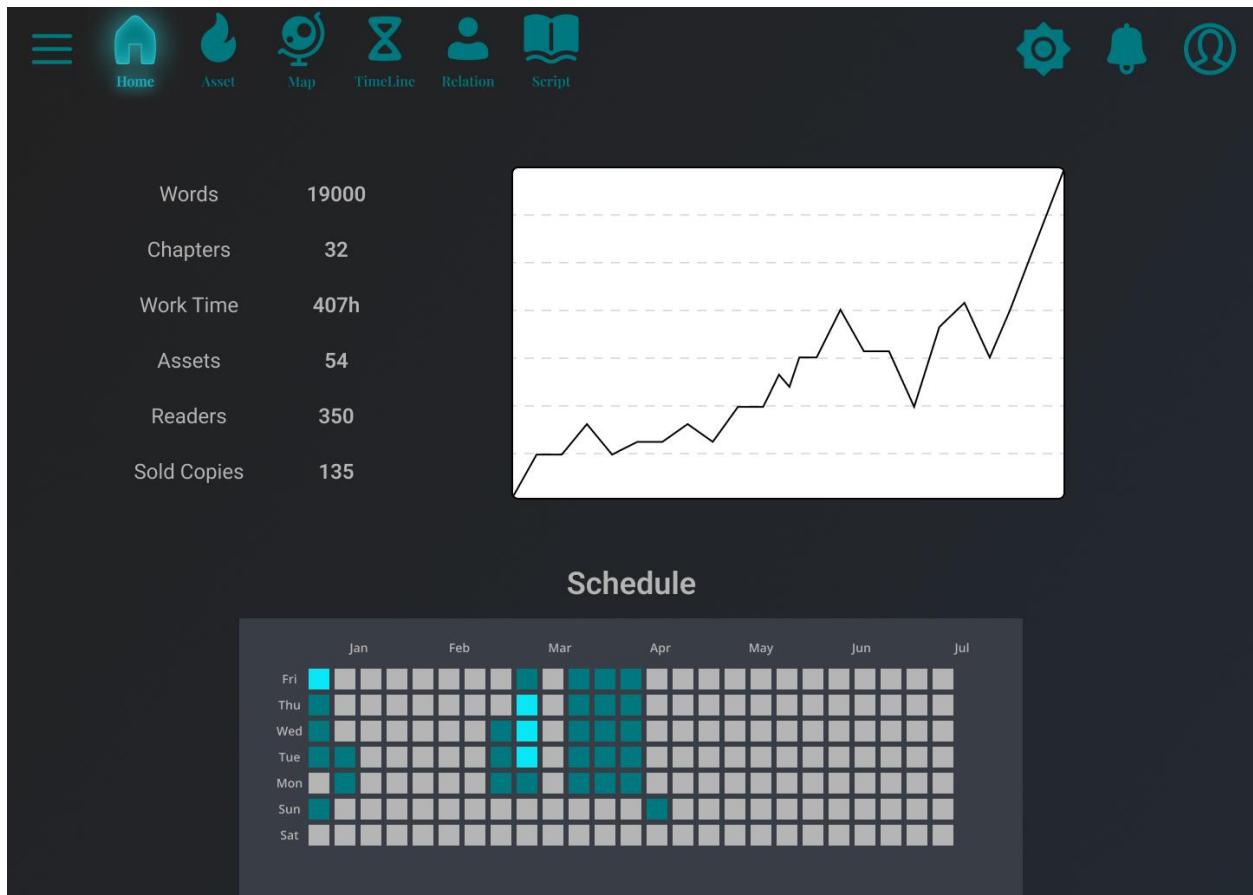
Alternatively, they can use this page to delete a project if they choose.



Delete Project

The user will be provided with a confirmation pop-up when they choose to delete a project. It helps to prevent unwanted loss of data by accidentally deleting a project.

If it becomes bothersome the user can choose to not be shown this pop-up again which they can reactivate on the settings page.



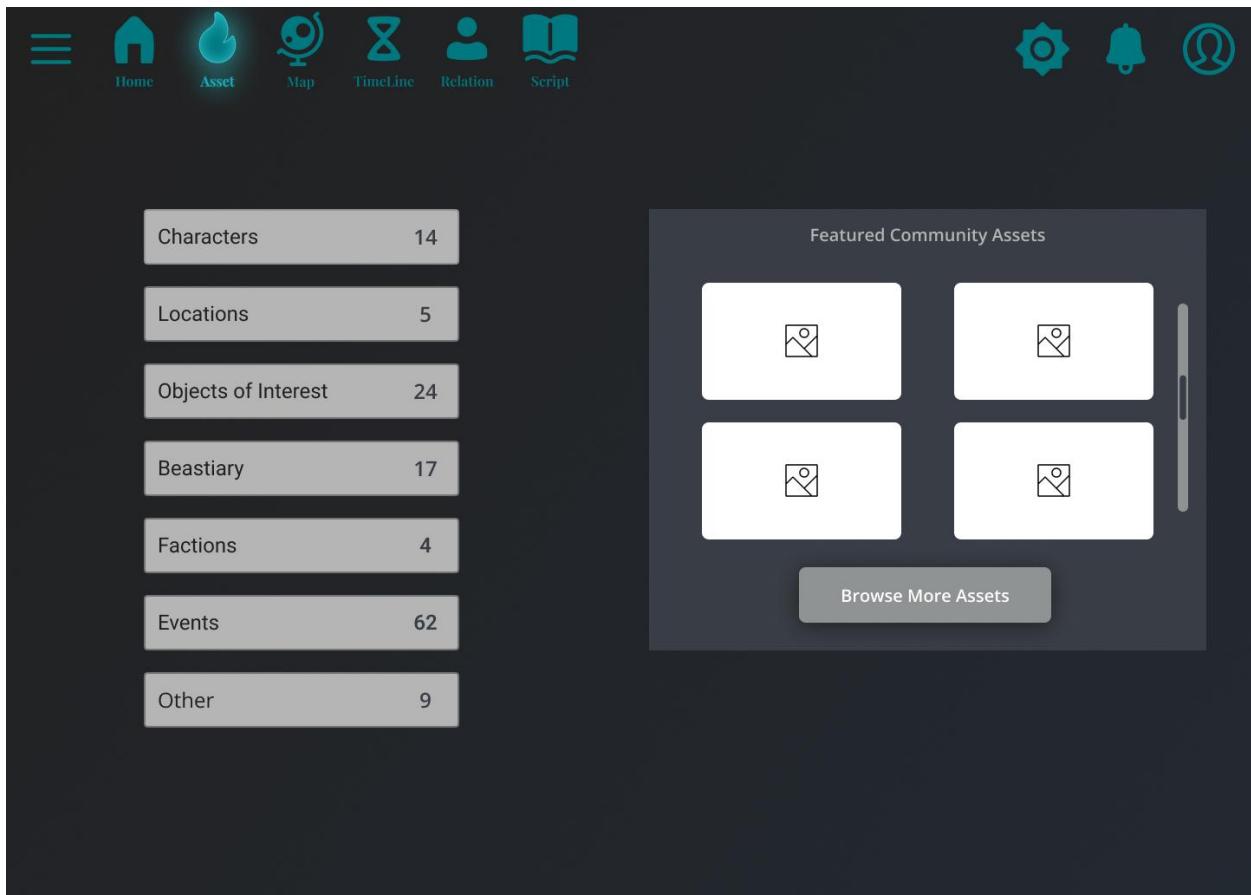
Project Panel

The project panel allows the user to work on their project. This page shows information about the project itself and the user's activity on this project.

This is calculated by time spent on the project, words and chapters written for the story and assets made.

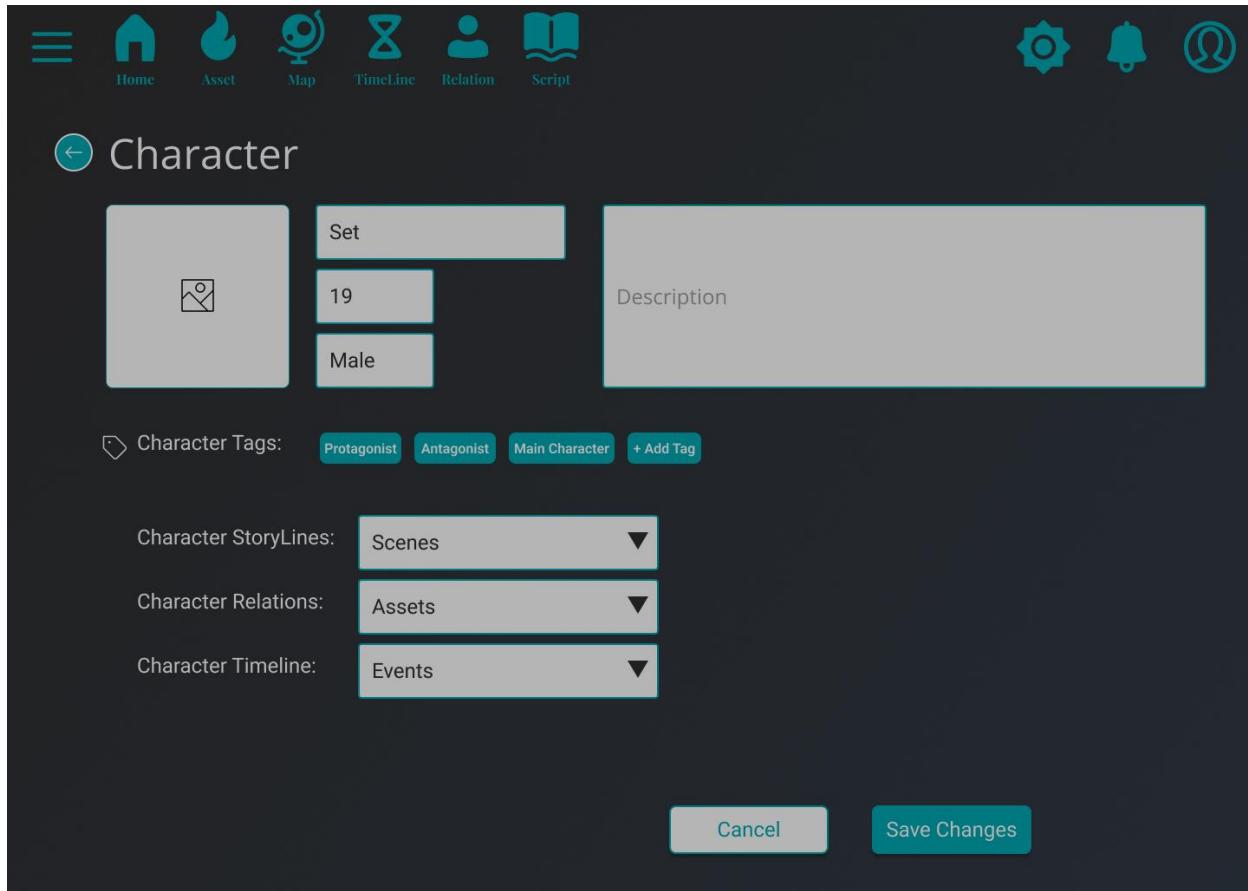
Finally, it shows analytics for the project's performance in the community like the number of readers and the number of people that bought the product.

It also allows the user to see their activity on the schedule and keep track of their work time for more productivity.



Asset Page

This page allows users to create and edit their story elements like characters, locations, creatures and story events. It also shows them the top community-made assets for them to download and use in their story.



Asset Creation

This page is accessed by users choosing to create an asset (in this case a character asset).

The first part of the asset creation is the information section, where the writer will provide some information about the asset and a brief description of them.

The second part is the tags section, where the asset will be set into groups based on their part in the story.

And the final part is the occurrence section, where the scenes in which the asset has participated in the story, the relationships they have with other assets and the events which included this asset will be mentioned.

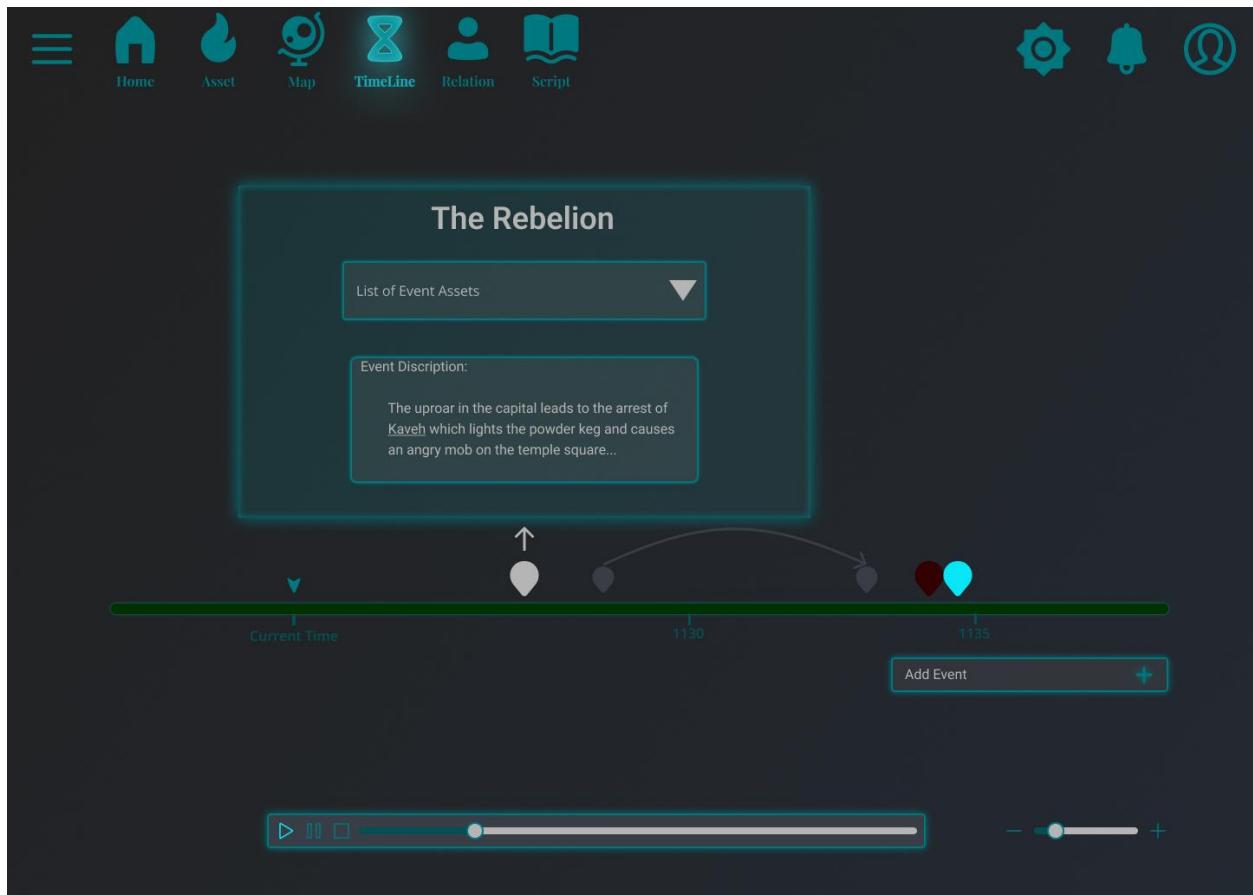
As the story progresses these assets will be updated and modified to accurately show their role in the story.



Story Map

The map feature allows the user to create a world atlas containing the story elements and their position in the world.

The user can draw a map using the palette provided (or a downloaded palette from the community) and then, they can drag and place assets on the map to set their location at a time.



TimeLine

The Timeline uses events to showcase what occurred in the story throughout the plot. The user can choose an event and place it on the timeline, or select an event on the timeline to show details about the event and a description of the event with a link to the scene or scenes describing that event.



Character Relations

The relation graph helps the user to visualise the connection between assets.

They can choose and drag assets from the left to the middle to place them on the graph and set the connecting edges to other assets manually or automatically based on the assets themselves.

Furthermore, they can view the changes in the relationships over the passage of time and plot.

Chapter 01: The hunt

List of Scene Assets ▾

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. A condimentum vitae sapien pellentesque. At auctor urna nunc id cursus. Consectetur purus ut faucibus pulvinar elementum integer enim neque volutpat. Commodo elit at imperdier dui accumsan. Cursus mattis molestie a iaculis at. Egestas purus viverra accumsan in nisi nisi scelerisque eu ultrices. Nibh ipsum consequat nisl vel pretium lectus. Laoreet id donec ultrices tincidunt arcu non sodales neque sodales. Nulla facilisi morbi tempus iaculis. Nunc eget lorem dolor sed viverra ipsum nunc aliquet. Eros in cursus turpis massa tincidunt dui. Nunc sed augue lacus viverra vitae congue eu consequat. Nisl tincidunt eget nullam non. Nam at lectus urna quis convallis convallis tellus id interdum. Non pulvinar neque laoreet suspendisse interdum. Consequat ac felis donec et odio pellentesque diam volutpat. Ornare aenean euismod elementum nisi quis eleifend quam.

Mi ipsum faucibus vitae aliquet nec ullamcorper sit amet. Tortor pretium viverra suspendisse potenti nullam ac tortor vitae purus. Nunc mi ipsum faucibus vitae aliquet. Posuere sollicitudin aliquam ultrices sagittis orci a scelerisque purus semper. Condimentum id venenatis a condimentum. Enim blandit volutpat maecenas volutpat blandit. Vivamus arcu felis bibendum ut. Volutpat sapien nec sagittis aliquam. Quis vel eros donec ac odio tempor orci dapibus ultrices. Neque aliquam vestibulum morbi blandit cursus risus at ultrices. Volutpat commodo sed egestas egestas fringilla phasellus. Suspendedisse ultrices gravida dictum fusce. Amet est placerat in egestas. At tellus at urna condimentum mattis pellentesque id. Tempor commodo ullamcorper a lacus vestibulum. Turpis egestas maecenas pharetra convallis posuere morbi leo. Mauris pharetra et ultrices neque ornare aenean euismod elementum nisi. Orci ac auctor augue mauris augue. Condimentum mattis pellentesque id nibh tortor id aliquet lectus.

B I U = = = ⌂ ⌃ ☰ ☱ ☲ ☳

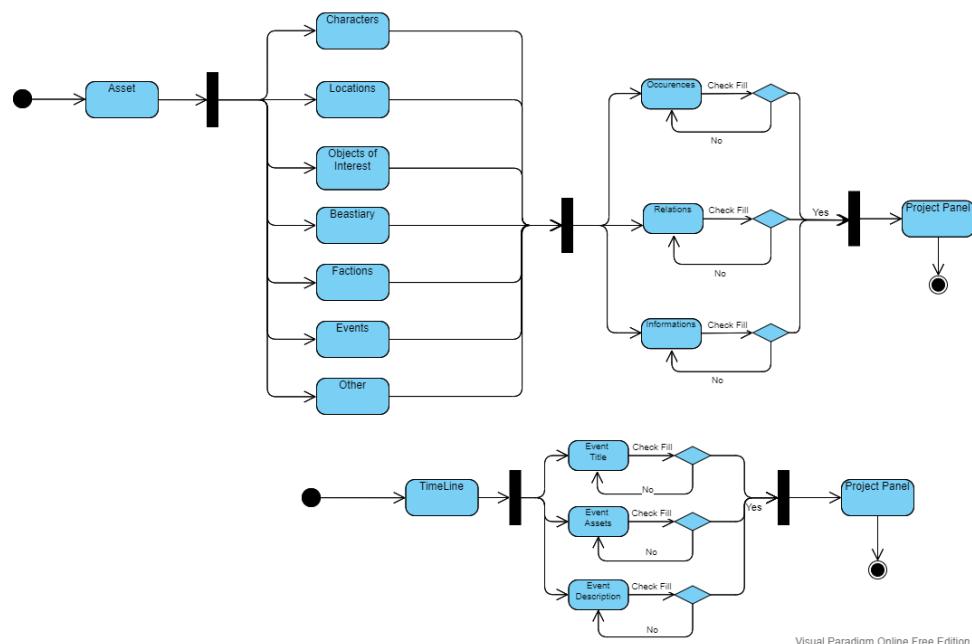
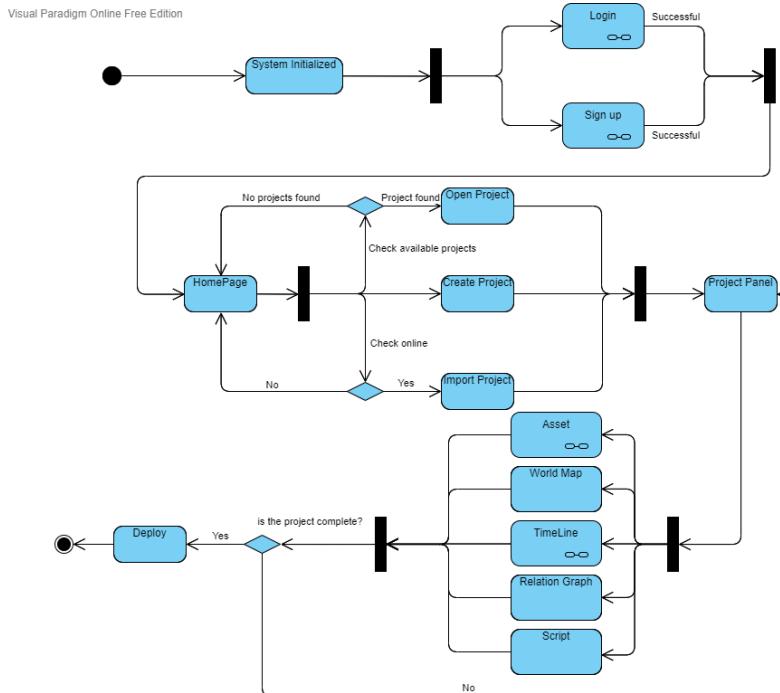
Story Script

On this page, the users can write their story while using the elements created beforehand.

Furthermore, the user can highlight the assets in the scenes to record their occurrence in the story for other features.

State Diagram

This diagram aims to help better understand the connection between the GUI pages and their interaction. So we can use this behavioural diagram to map out the GUI from start to end of the session.



Structural Diagrams

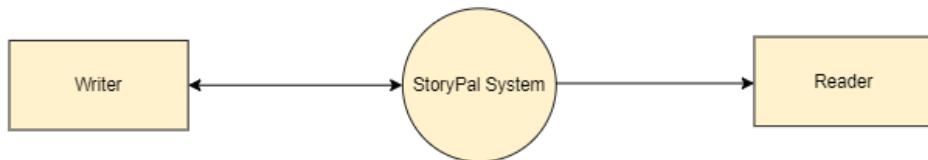
In this section we focus on the inner working of the software parts.

We explore what are they and where and how they interact with each other.

We start with a Data Flow Diagram then we move on to Class, Component and Deployment Diagrams.

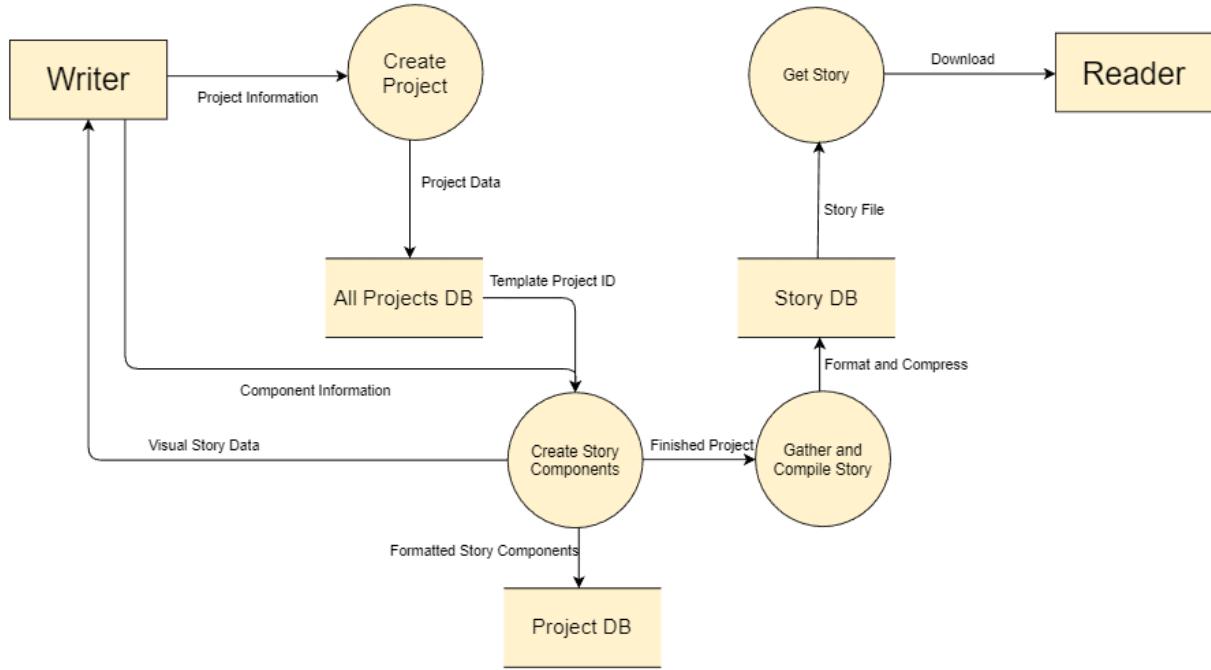
Data Flow Diagram (DFD)

Context Diagram

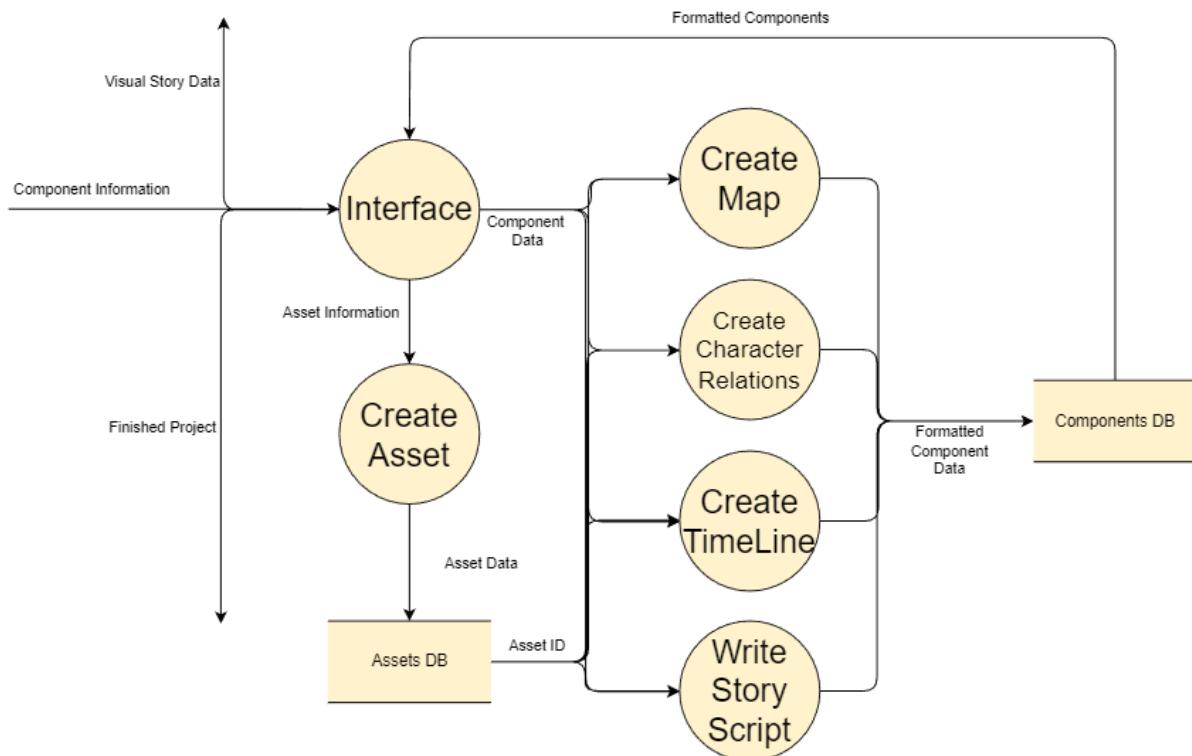




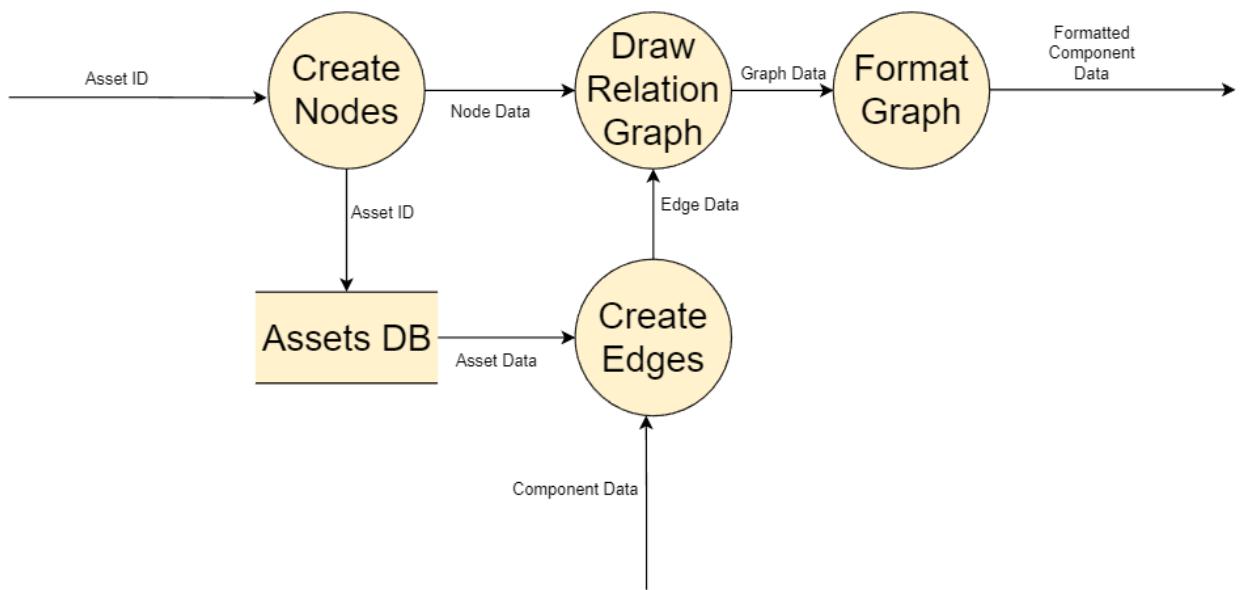
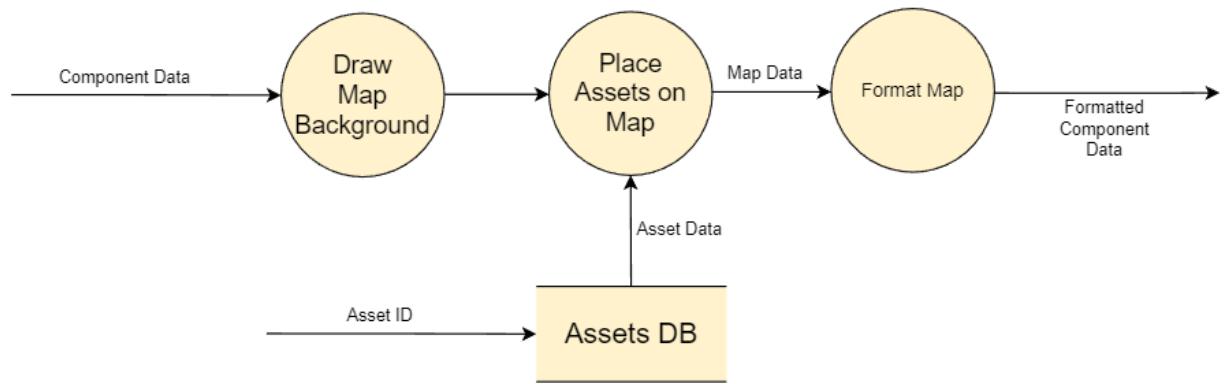
Lvl 0 diagram



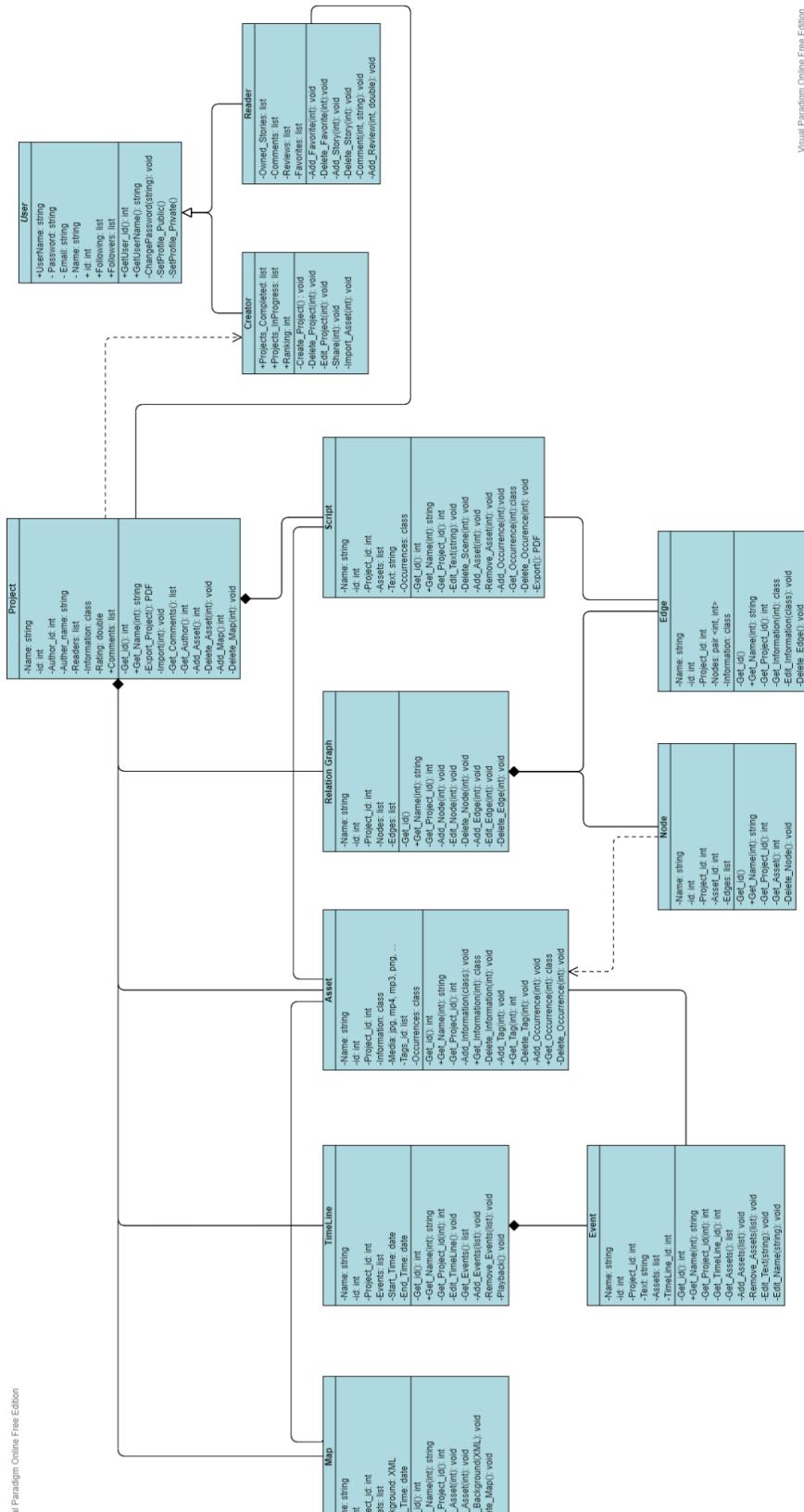
Lvl 1 Diagram



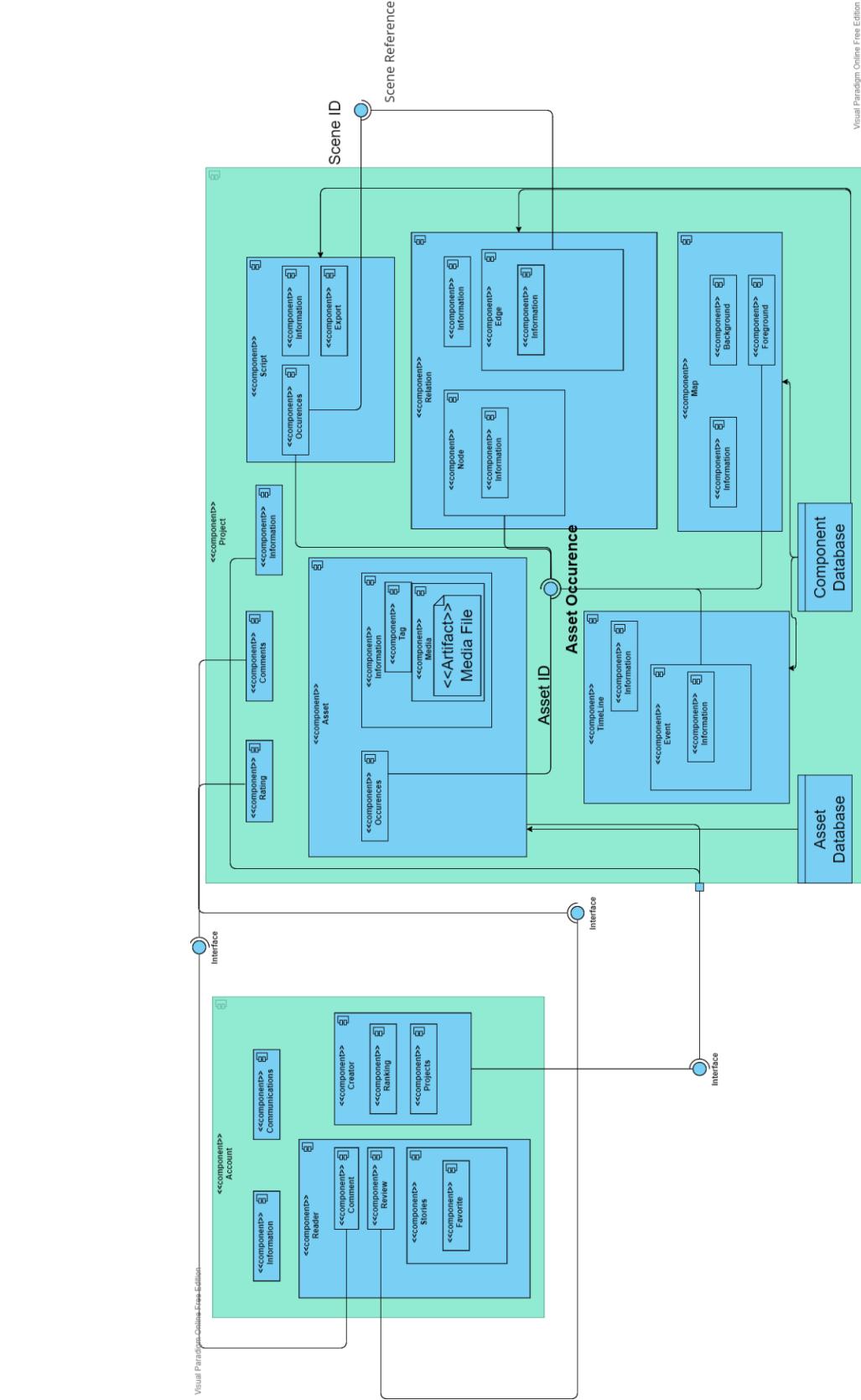
Lvl 2 diagrams (Map Creation and Character Relation)



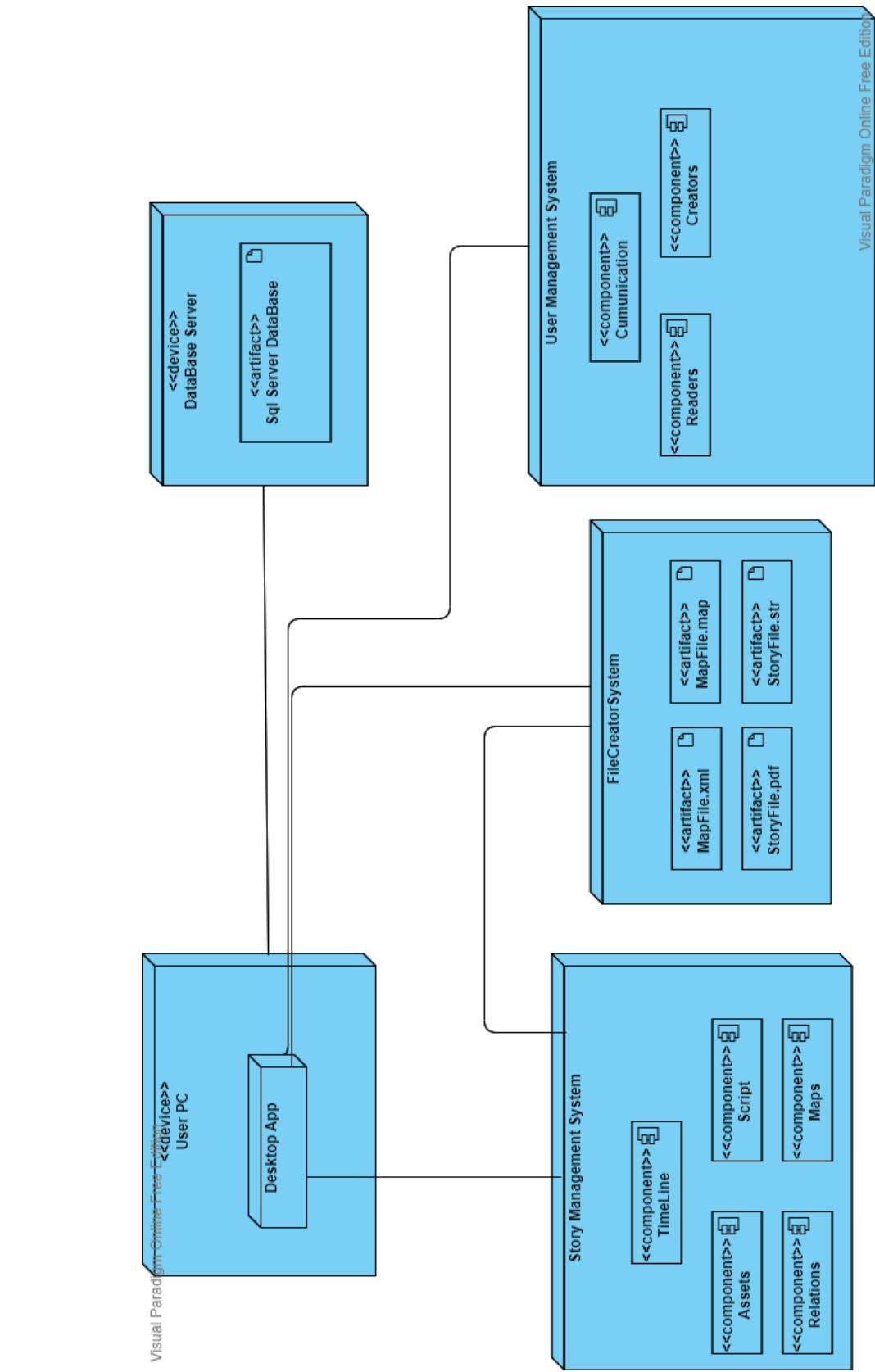
Class Diagram



Component Diagram



Deployment Diagram



All of the Diagrams are available with better resolution at these links:

Data flow diagram:

<https://online.visual-paradigm.com/share.jsp?id=323033383336342d31#diagram:workspace=spvudtqj&proj=0&id=1>

Class Diagram:

<https://online.visual-paradigm.com/share.jsp?id=313937313732332d33>

Component Diagram:

<https://online.visual-paradigm.com/share.jsp?id=313933393334352d35#diagram:workspace=gzkyabfk&proj=0&id=5>

Deployment Diagram:

<https://online.visual-paradigm.com/share.jsp?id=323036383634302d34>

Figma MVP prototype:

<https://www.figma.com/file/P1NcvRN7IcyI8VabU64YX/Figma-mvp?node-id=0%3A1>



What's Next?

This was a fantastic endeavour in which our team learned a lot and gained much-needed experience. But we aren't done.

Our team is filled with people eager to learn and see this app implemented.

We plan to familiarize ourselves with software programming and start working on this app in the summer of 2022.

Based on the team members' expertise in C# and python and Django for the web-based version. Our hopeful estimation is 6 months of development for both versions to be functional.

After that, we can focus on the implementation much needed extra features that were cut out based on needs and difficulty of execution. A functioning purchase system and a well-developed communication channel between writers and publishers are much needed and are intended to be developed in the first major patch.

And finally, after that, we plan to branch out. Foster, our community of writers to help us grow our app and expand into other writing genres.

Thank you to Anita Seihoon, Professor Hossein Nourikhah, and many others for their help.



**And thank YOU for your
time and attention!**