UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Puzzle Generator System Requirements Document

Team Prometheus

23 July 2020

Clients: Mark Anthony and Francios

Name	Student Number	**Email**
Yuval Langa	u18174142	u18174142@tuks.co.za
Tsholofelo Gomba	u17391718	u17391718@tuks.co.za
Tapiwa Mazibuko	u18203541	u18203541@tuks.co.za
Charlotte Jacobs	u15165622	u15165622@tuks.co.za
Jaynill Gopal	u15175295	u15175295@tuks.co.za

Contents

1 Introduction

1.1 Product Name

Prometheus Puzzles

1.2 Purpose of the Software System:

The Puzzles Generator website aims to efficiently provide an interface to users to generate puzzles through the use of AI or create them manually. The AI based system will make use of a genetic algorithm.

1.3 Scope:

This system will aim to allows for the user to create a puzzle based on a pattern/style of choice as well as with different number of pieces per puzzle.

The system aims to allow users to share creations on the website and rate creations made by other users. A user should finally be able to use the system to generate a printable 3D model of a puzzle.

1.4 Business need:

The business need of this system is to automate the process of creating puzzles. The website design must be user friendly and appealing in order to attract users to interact with the system. Added to this, the steps taken to create the puzzle must be easy to follow and the process efficient.

1.5 Overview:

Puzzles have been a hobby of many people over the centuries, coming in various forms which have various ways of solving. Puzzles have played a role in people's problem solving skills. Puzzle generators have allowed for more puzzles to be created, using various techniques to create interesting and challenging puzzles.

The system involves the creation of 3-Dimensional puzzles (manually and from the use of AI), testing of puzzles, sharing and rating of puzzles by other users, as well as the ability of downloading 3D printable files.

2 User Characteristics

The general user of the website is someone proficient in English so they can easily navigate the website and perform tasks as instructed. The user also needs to have an understanding of how to use technology, even though we aim to make our application very user friendly.
As such, we classified our users according to the following categories:

- Puzzle Enthusiasts - these can be users of any age, be it little kids, teenagers or adults who like playing with puzzles.
- Parents - some parents may want to make puzzles for their children to play with.
- Educational Users - these can be teachers who want to teach children with conditions like autism, motor skills.

3 Functional Requirements

3.1 Use Cases

- UC1: Register
- UC2: Login
- UC3: Logout
- UC4: View Own Puzzles
- UC5: View Rated Puzzles
- UC6: Update Name
- UC7: Update Username
- UC8: Reset Password
- UC9: View Puzzles
- UC10: Rate Puzzles
- UC11: Update Puzzle Ratings
- UC12: Share Created Puzzles
- UC13: Search Puzzle
- UC14: Export Printable 3D File
- UC15: Manually Create Puzzle
- UC16: Manhattan Puzzle Creation

- UC17: Euclidean Puzzle Creation
- UC18: Generate Puzzle Using AI
- UC19: Select Puzzle
- UC20: Test Puzzle

3.2 User Stories

1 As a User I want to create an account so that I can create my own puzzles

2 As a User I want to have a manual generation option so that I can create my own puzzle designs

3 As a User I want to have a AI generation option so that I can tell it to create a puzzle based on certain options.

4 As a User I want to share my puzzles creations so that I can other users can solve and rate them

5 As a User I want to stop sharing my puzzles creations so that I can other users wont be able to print my creations

6 As a User I want to rate created puzzles so that other users can be encourage to work on creating more puzzles

7 As a User I want to update the rating on puzzles so that other users can see how I feel about their puzzles

8 As a User I want to view all shared puzzles so that I can see what other users are sharing

9 As a User I want my own profile page so that I can manage my personal ratings and creations

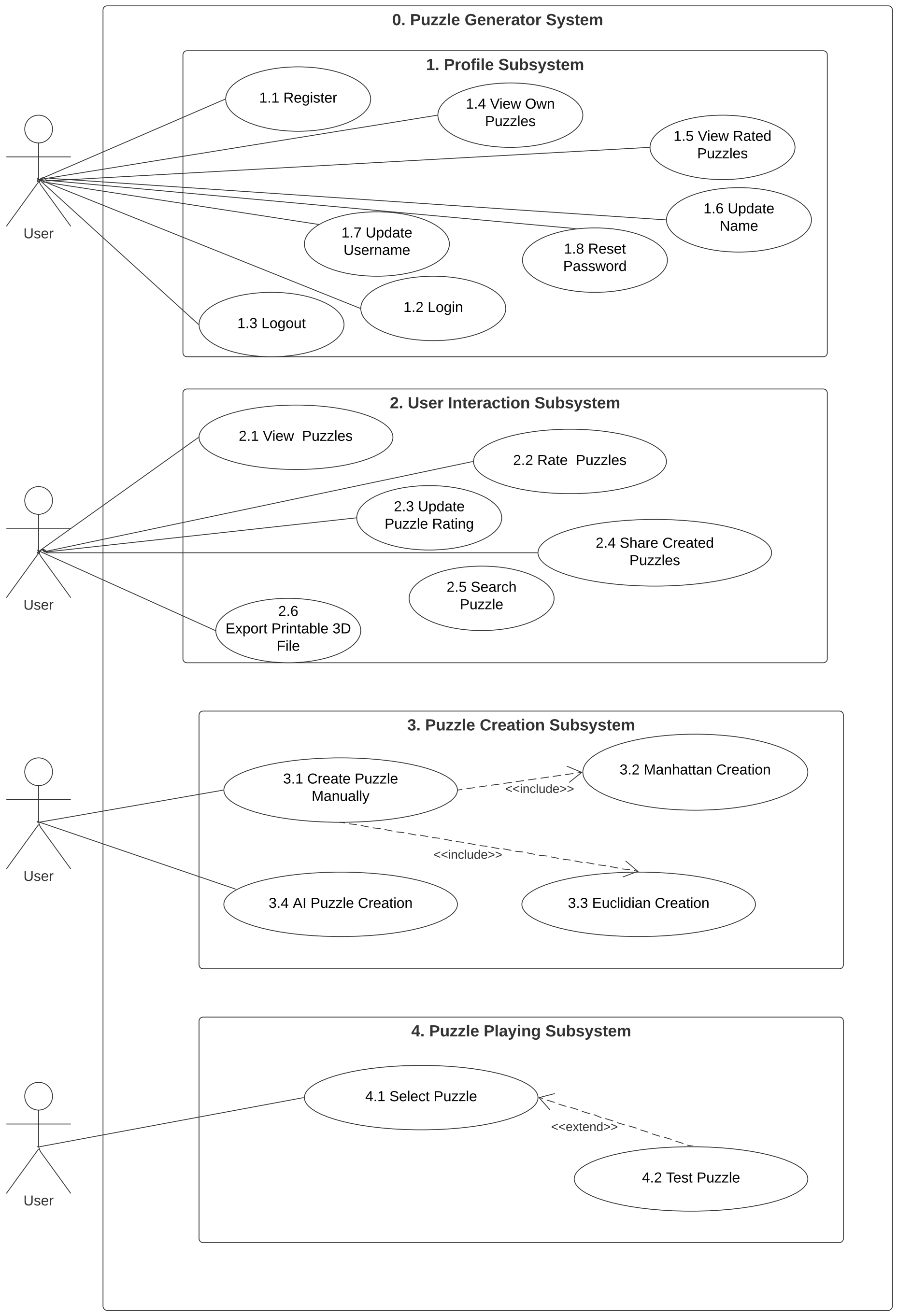
3.2.1 Use Case Diagram Description

Profile Subsystem
A user can use the register use case to create a new account, login as a registered user, and logout of his/her account. Once the user has a profile, the user can view a list of the puzzles that s/he has created and rated.

User Interaction Subsystem
This subsystem aims to highlight how the user interacts with created puzzles on the website. The user can view created puzzles, rate these puzzles, share their own puzzle creations and export puzzles to printable 3D files.

Puzzle Creation Subsystem

Figure 1: Use Case Diagram



The user can uses this subsystem during puzzle creation. The user selects a desired shape from which they can decide to create the puzzle manually or use the AI to generate the puzzle. If a user decides on manual puzzle creation, they may upload an image for the puzzle

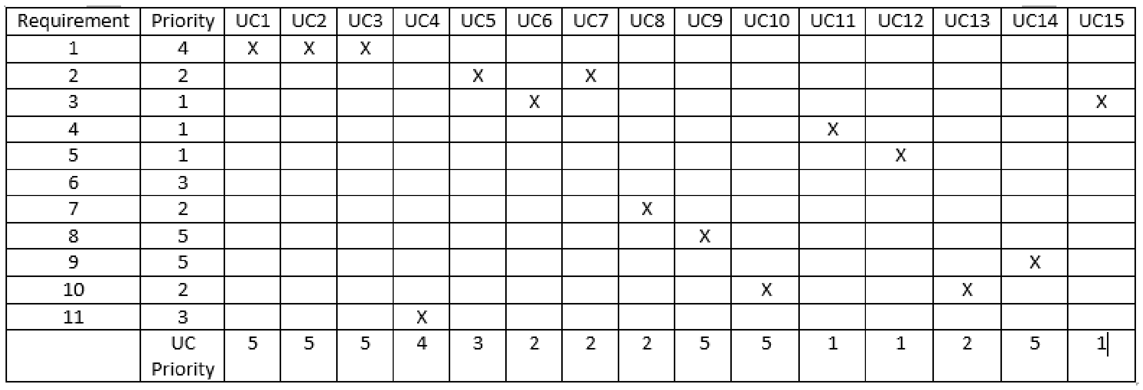
Puzzle Playing Subsystem
The user interacts with this subsystem when playing a puzzle. The user selects a desired puzzle and tests it (plays it).

3.3 Requirements

- R1: The System must allow the user to register, and login to, a user profile.
- R2: The system must allow the user to rate puzzles.
- R3: The system must allow the user to view or play puzzles.
- R4: The system must allow the user to create a puzzle.
- R5: The system must be able to generate a puzzle through the use of Artificial Intelligence.
- R6: The system must be hosted on a web server.
- R7: The system must be able to export a puzzle to a 3-dimensional printable file.
- R8:The system must be able to store puzzles that have been created.
- R9: The system must be allow the user to choose the initial shape of the puzzle.
- R10: The puzzle must be able to upload an image to create a puzzle from.
- R11: The system must be able to store user profiles.
- R12: The system must be able to search and filter the puzzles.
- R13: The system must be able to use Euclidean distance ans Manhattan distance to make the puzzle.
- R14: The system must be able to update the username and the name of the user.

3.4 Subsystems

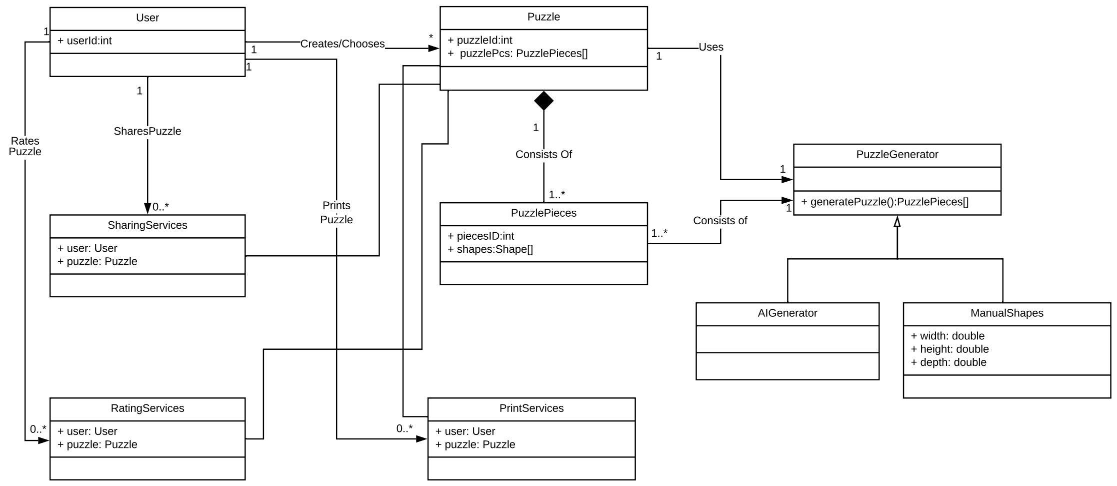
3.5 Trace-ability Matrix



Requirement	Priority	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15
1	4	X	X	X												
2	2					X		X								
3	1						X									X
4	1											X				
5	1												X			
6	3															
7	2								X							
8	5									X						
9	5														X	
10	2										X			X		
11	3				X											
	UC Priority	5	5	5	4	3	2	2	2	5	5	1	1	2	5	1

4 Domain Model

Below is our domain model. The AIGenerator class will make use of pattern recognition in AI.



4.1 Description of classes

4.1.1 User

The User class will be used for user management. Each user can create and choose as many puzzles as they want. Each user can also rate, share and print as many puzzles as they want hence the link between the User class and SharingServices, RatingServices and PrintServices classes.

4.1.2 Puzzle

The Puzzle class is used to maintain the different puzzles. The puzzle consists of 1 to many different puzzle pieces. When AI is used, i.e. the puzzle is not generated manually, the puzzle class makes use of the puzzle generator class which in turn uses pattern recognition to create the different puzzle pieces. The puzzle class is linked to the user class as the user interacts with the puzzle and each puzzle is owned by a user. The Puzzle class is also linked to the SharingServices, RatingServices and PrintServices classes as each of these services is linked to a certain puzzle.

4.1.3 PuzzleGenerator

The PuzzleGenerator will use AI in specific pattern recognition to generate the puzzle pieces for the Puzzle class.

4.1.4 Shape

The Shape class is used to get the specific shape of the puzzle pieces. Each puzzle piece can be made up of one to many different shapes.

4.1.5 SharingServices

The SharingServices class is responsible for sharing puzzles. The SharingServices class is linked to the user and puzzle classes as this is needed when using the service.

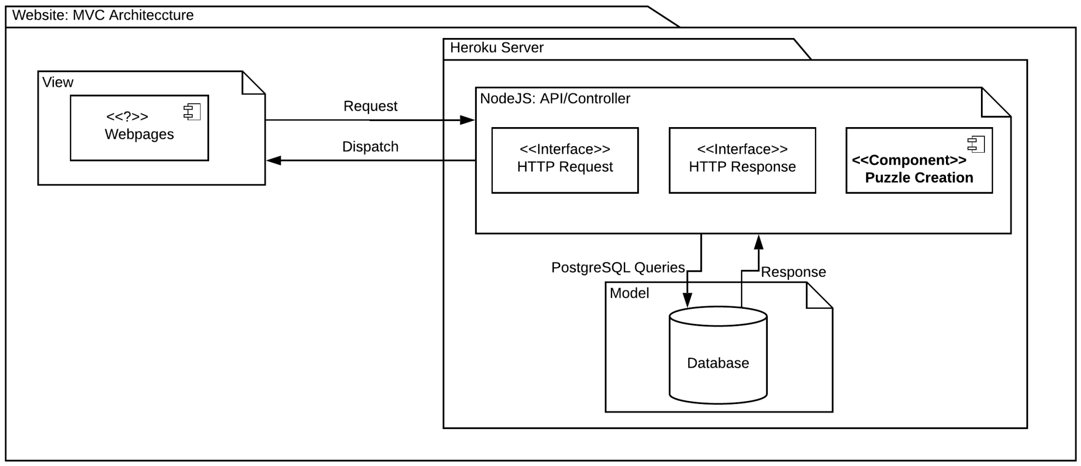
4.1.6 RatingServices

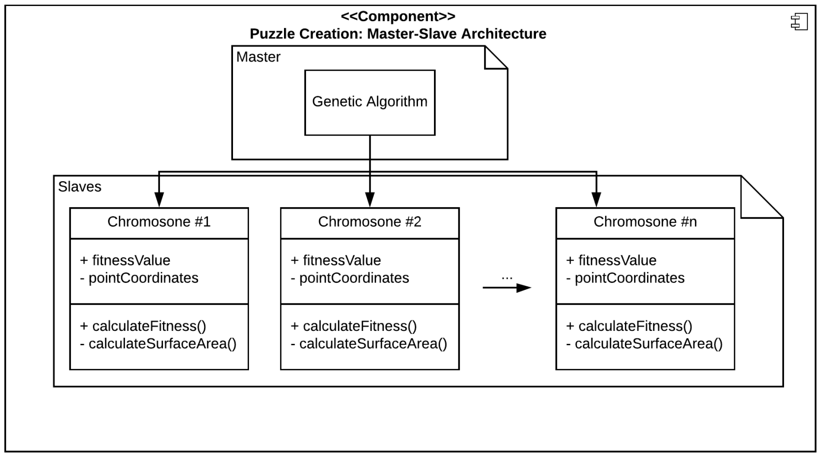
The RatingServices class is responsible for rating puzzles. The RatingServices class is linked to the user and puzzle classes as this is needed when using the service.

4.1.7 PrintServices

The PrintServices class is responsible for printing puzzles. The PrintServices class is linked to the user and puzzle classes as this is needed when using the service.

5 Architectural structural design requirements





5.1 AI Puzzle Creation

We are making use of the Master-Slave architecture pattern to implement a Genetic Algorithm to tackle this portion of the system, structured as follows:

- Slave - the chromosomes represent the Slaves, as they are manipulated by the Master
- Master - the algorithm itself is the Master, as it manipulates the data in the slaves

The user makes a request to the API with some initial data (through the website in the browser). The API will then execute the Genetic Algorithm and then return a JSON object containing the vertex points, which will then be used to render the puzzle onto the user's canvas.

5.2 Front End Implementation

We are making use of the MVC architectural pattern to implement this portion of the system, structured as follows:

- Model - we are making use of a Postgres database to store data such as user and puzzle details
- View - we are using the Angular framework to create components representing individual pages to be rendered in the browser.
- Controller - we are making use of an API written using NodeJS and the Sequelize ORM to query the database.

The user interacts with the website in the browser (view using Angular). The request generated from the user interaction is sent to the API (controller using NodeJS), which will manipulate the data in some way and render the results to the user in the view once more. Some requests might need database access (model using Postgres). Appropriate data is then retrieved or saved as needed.

5.3 Technologies

- Hosting
  Heroku's free tier is used to host the website, allowing the user to access and create puzzles.
- Server
  NodeJs using Express framework is used to host the API on the server side, which is used to allow interaction between the user and database as well as hosting the automatic puzzle creation.
- Web page
  The web page will allow users to access and to have puzzles generated, which they can then download and save as a 3D printable file.
- Database
  PosgreSQL will be used to store data required for the users, there are three main tables: Users, Puzzles and PuzzleRating
- Frameworks
  - AngularJS will be used so that the web page will have a responsive design without needing to load a new page.
  - Sequilize will be used to allow for querying the PostgreSQL database.

6 Quality requirements

6.1 Usability

Users with a basic technological skill level must be able to freely create puzzles and use the website.

- This requirement is addressed by the layout and design of the user interface.
- The front end of the website is easy to navigate and visually pleasing for a better user experience.
- The front end uses an angular framework to serve web pages.

6.2 Performance

The AI should be able to generate a new puzzle between 30-60 seconds. Due to the real time nature of the website.

- This requirement is very important and will be addressed by a combination of architectural design patterns.
- The performance of the of the manual puzzle generation is less computationally taxing and will be a part of the MVC design pattern.
- The AI puzzle generation will be efficient because of the genetic algorithm using the master slave design pattern which will allow the same algorithm to run efficiently with different inputs.

6.3 Reliability

The website should be able to handle traffic from hundreds of users at any given time without crashing or errors.

- This requirement is very important and will be addressed by both the web server and back-end architectural design.
- The website uses Heroku web server which is a highly reliable website hosting service.
- The MVC pattern is being used with an angular framework and node js to control the requests made to the website and database.

6.4 Availability

The website should be up 99.9% of the time.

- This requirement will be addressed by both the web server and database.

- The website uses Heroku web server which is a highly reliable website hosting service.
- The website uses postgre which is also available at all times.

6.5 Cost

The software and libraries used must be free/open source.

- This requirement will be addressed by using open source libraries and software.
- The website uses the free tier of the Heroku web server.
- The website uses open source architectural technologies such as angular and node express.

6.6 Security

The system requires security measures to protect user data.

- This requirement is necessary because the website will store sensitive user information.
- The MVC architecture allows for separation of API request to allow more security.
- The website will also protect user information by using password encryption and database management tools such as sequelize.

7 Coding standards document

7.1 Server component

NodeJs Express applications has it's own folder structure. The folder structure of the server is as follows:

- Front-end files are listed under a directory called **Public**. The structure of the directory is as follows:
  - /stylesheets
  - /javascript
  - /images
  - index.html
- Back-end related JavaScript files are listed under the directory **Router**, after which the routing of the API call are in separate files:
  - api.js

  - user.js
  - puzzle.js

Unit testing of back-end the API is tested through Mocha/Chai frameworks. The testing file is listed as test.js in the root folder of the application base.

7.2 Angular component

Angular by design has its own folder structure which we maintained during the creation of our front-end.

- Every website page is a component (e.g. landing page, login page etc) are stored within the pages folder. Please follow the link to see the structure on our Github page
- The shared navbar is in the root app folder for easy access to components needing it
- API calls are defined in a service stored under the services folder
- Models are defined and stored under the models folder
- All images can be found in the assets folder
- All components have a .css, .ts and .html file. Component specific code is defined in the relevant files while global styles are defined in the app css, ts and html files.

7.3 Node API component

We made use of NodeJS to write our API. Calls to the database are made using the Sequelize ORM. Our file structure is as follows:

- Please follow this link to visit our Github page to see the structure
- config folder - it contains the database configurations
- routes folder - it contains files which define the routes for particular endpoints. A typical endpoint route is "api/users/createUser", and the folder therefore contains a users.js file contain routes concerned with the user which in case would be createUser
- models folder - this folder allows the definition of models as defined in the database as per Sequelize standards. An example would be the User class defining the fields which are found in the users database table. A model is referenced when doing database operations

8 User manual

8.1 Server setup

8.1.1 Prerequisites

Ensure that you have the following installed on your system.

1. NodeJS version 3.0+
2. npm
3. PostgreSQL

8.1.2 Installation on localhost

In order to use the server and load files to be hosted on the website the following needs to be done.

1. Open terminal to the directory that the
2. Enter the following command to install dependencies: **npm install**
3. Once the dependencies have been downloaded, you are now ready to run the server.
4. To run the server enter the following command into your terminal in the root folder of your app: **node start**.
5. You are now ready to access your server at **localhost:3200** in your browser.

8.1.3 Customizing your server

1. Changing server's default port
   If you wish to alter the default server port on localhost, you may do so by editing the file:**/bin/www** and locating the variable `PORT`

8.1.4 Adding your web pages to be sent

1. All additional web pages and subsequent CSS and JavaScript files intended for the use of the user should be placed in /public /public/stylessheets /public/js
2. to access the API the following routes are used:
   - /api/user/login
   - /api/user/createUser
   - /api/user/resetPassword
   - /api/puzzle/create
   - /api/puzzle/rate
   - /api/puzzle/view
   - /api/puzzle/viewRatings

8.2 Accessing the website

In order to access the website you can follow the following link: https://prometheuspuzzles.herokuapp.com/. From here you will be greeted with our homepage, where you have the options to login or sign up, or simply browse through the available puzzles to view. ¡¡Insert homepage here¿¿

8.3 Website features

8.3.1 Splash page

The splash page is used as a page where users can sign up, login or view some of the created puzzles as a preview.

8.3.2 Sign up

The user will be able to click on the sign up button on the splash page where they will be redirected to a sign up form. Once a user has signed up, they can log in using their personal details.

8.3.3 Login

The user can click on the login button on the splash which will redirect them to the login form. Once a user has logged in they can create, rate and view puzzles and also view their own profile page.

8.3.4 View

The view page shows all created puzzles by other users. By clicking on a specific puzzle, they can rate or solve the puzzle or just get more information regarding that puzzle.

8.3.5 Rate

When a user clicks on the rate button on the view page the user will be redirected to a rate form. This way users can rate each others' puzzles.

8.3.6 Create

A logged in user can go to the create tab and start creating their own puzzle. They will be given a canvas with a square, when clicking on the square the square will be divided at that point. The amount of times the user click on the square, is the amount of pieces the puzzle will have.

8.4 To be added

Screenshots of all working components will be added before demo 2

8.4.1

Capture1.png

Capture2.png

Capture3.png

8.4.2

Domain.png