Software Engineering CSC648/848

Section 4 | Team 2

Artemis

3/18/2022

Brain Adams	badams2@mail.sfsu.edu	Team Lead
Joseph Kois	jkois@mail.sfsu.edu	GitHub Master
Brandon Cruz-Youll	bcruz5@mail.sfsu.edu	Scrum Master
Sherman Wong	swong42@mail.sfsu.edu	Backend Lead
Arslan Alimov	aalimov@mail.sfsu.edu	Backend Support
Jade Simien	jsimien@sfsu.edu	Frontend Lead

History Table:

3/1/22	Evaluated M2 for work needed to complete target. Divided work up and assigned tasks to team members.
3/8/22	Short stand up for progress on individual team members piece of the project. Created some models for the backend.
3/15/22	Went over remaining M2 tasks and team workloads. Front-end met and discussed refining UI/UX flow and elements needed. Back-end discussed finishing preparations for vertical sw prototype

1. Data Definitions V2

Name	Definition	Usage
Bug	Behavior that is unexpected or unintended	Tickets can cite known bugs to fix. Developers will be able to associate known bugs with tickets.
Priorities:	Scale of urgency that bug needs to be fixed	To easily ascertain the required urgency the ticket needs attending to, the bug priorities will be easily visible for quickly ascertaining which tickets need to be worked on
Critical	Highest priority bug to be assigned and worked on with urgency. Color: Red	
High	Bug that needs to be assigned and worked on as soon as possible. Color:	
Medium	Bug that should be assigned and worked on sooner rather than later, but not mission critical. Color: Yellow	
Low	Lowest priority bug to be assigned and worked on time willing. Color: Green	
Ticket	A single task that needs to be done	There will be lists of tickets that developers

Bug Title	Short description of bug to be displayed on ticket and Recently Viewed Tickets	can be assigned to. These tickets can be closed once the task has been completed. Tickets can include tasks like bug fixes, maintenance and refactoring. Tickets will start in the Issue Pool and be moveable to user-defined columns in the Workspace of the project. Tickets will be searchable.
Bug description	Full description of problem, viewable by clicking on ticket.	
Priority Status	Bug priority to be displayed on left side of ticket, represented by colors	
Assignment Icon	Developer assignment to be displayed on right side of ticket. Will be grayed out if unassigned, and filled in if assigned	
Comments	Tickets will have logged comments with date/time/developer to log status/updates/and any issues that are blocking the completion and closing of the ticket. Stack traces may be placed	
Timestamp	System generated when user comments on ticket	
Close	The completion of a task, to mark a ticket (task) as being complete	Developers will be able to close their tickets after the task is completed.

Viewable	Users will be able to click on closed tickets and view all the closed tickets of the project	
Workspace Boards	A collection of project boards, and potentially teams that belong to a particular organization. Workspaces contain all the information for a specific team or project, and are containers for tickets to be created and closed in. Project boards	Workspaces serve as a container to organize all the data for any given project or team.
Team Privileges	A collection of users who have specific permissions Set by organization along with job titles. Varying amount of privileges that allow: creating tickets, closing tickets, moving tickets, assigning tickets to developer, assigning tickets to team Workspace.	Teams are used to organize users and what they can alter within a given Workspace. If you belong to a developer team you can close tickets, if you are a part of a QA team you can create tickets etc.
Organization	An entity that can have multiple Workspaces and teams.	Organizations can provide a way for owners to maintain various Workspaces.

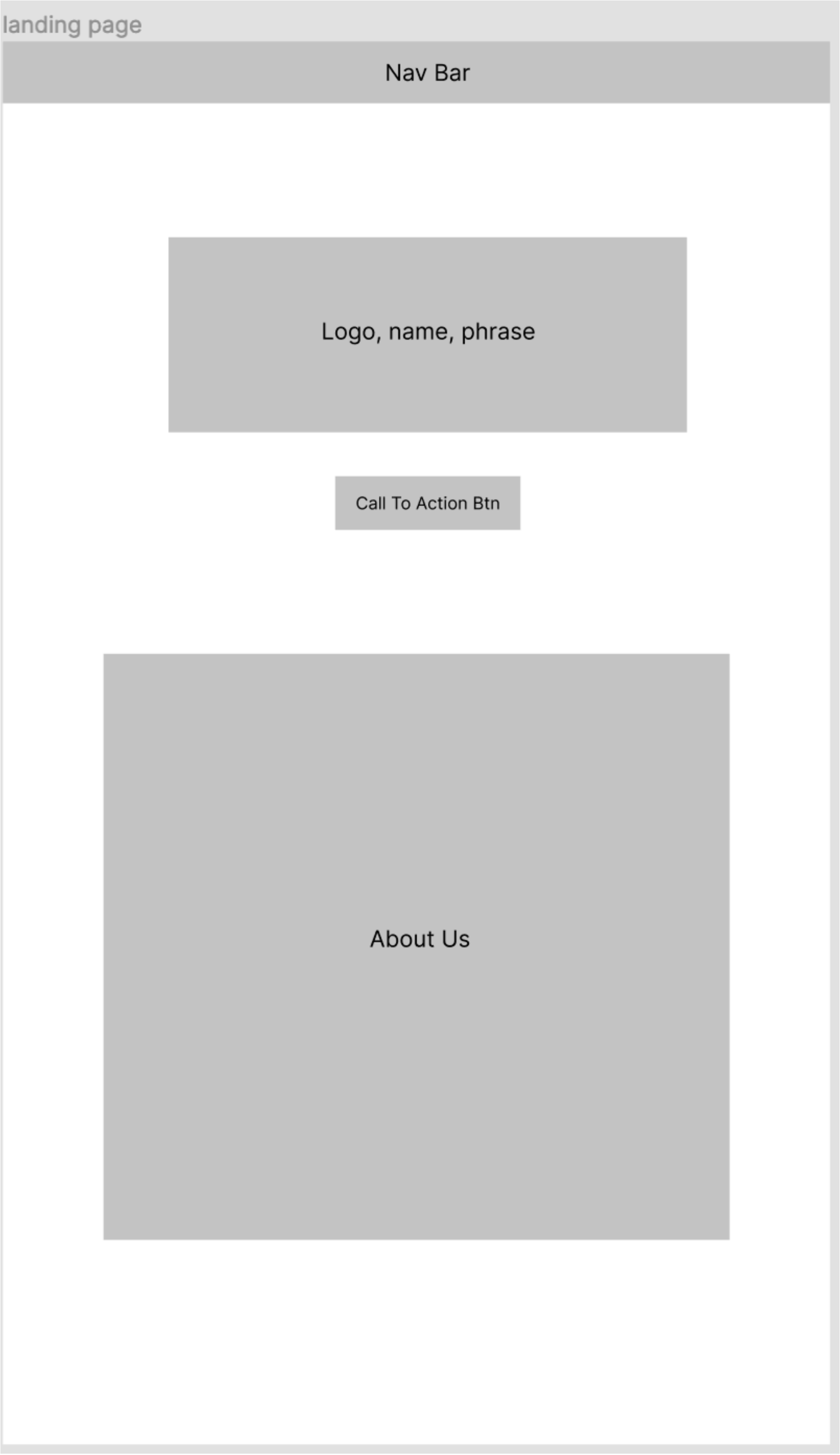
2. Functional Requirements V2

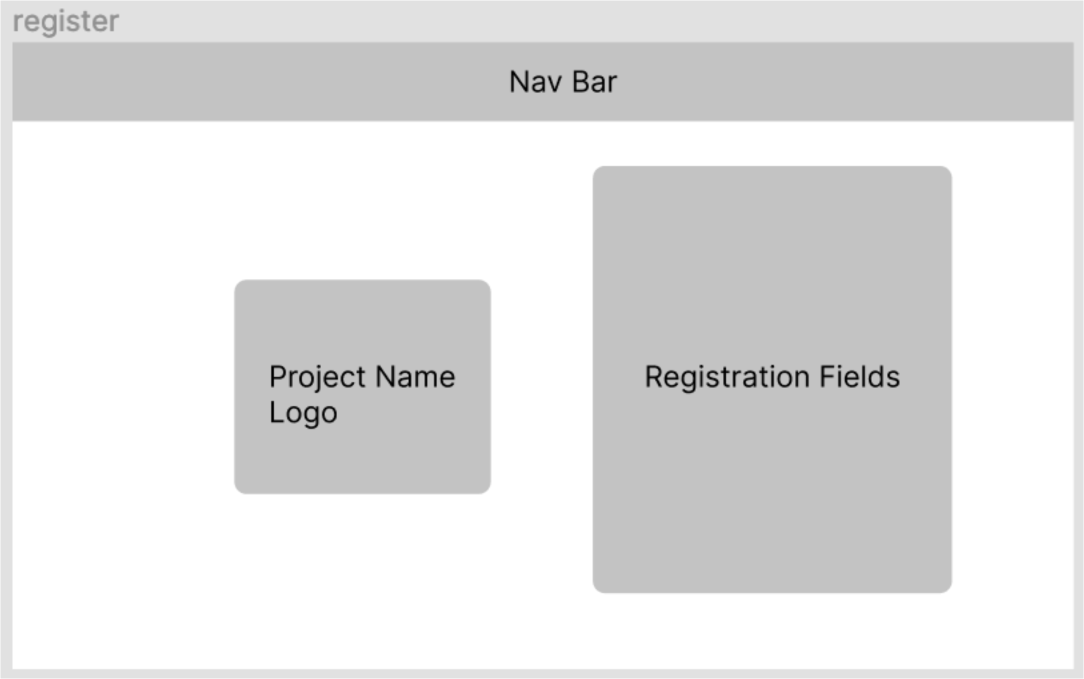
Number	Priority	Requirement	Details
1	1	**Tickets**	Includes priority status, assignment status, bug title, bug description and comments.
1	1	Ability to create and modify tickets	Click + to add a new ticket. Click on ticket to view/modify: bug title, bug description, developer assignment.
1.1	1	Tickets will have bug title, bug description, bug priority, user assignment, comments	These should be modifiable
2	1	Users should be able to assign tickets	
2.1	2	Based on permissions, users will be able to assign tickets	
3	1	Users should be able to mark tickets as closed	Once completed, ticket will be closed out with final comments, date/ time/and user who closed it out.
3.1		*see closed tickets are viewable*	
4	2	Ability to attach stack traces and descriptions to tickets	

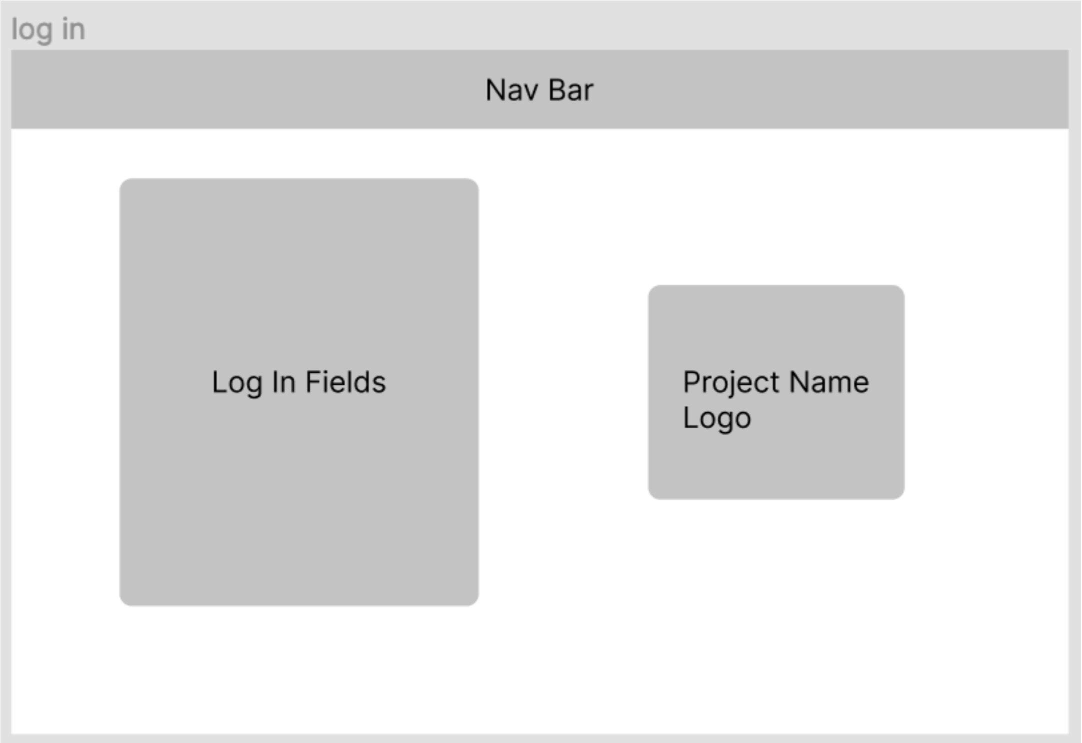
5	1	Closed tickets are viewable	Users should be able to click on ‘closed tickets’ to view all closed out tickets
2	1	**Workspace**	
1	1	Create, Destroy Workspaces	Based on user permissions
1.1	1	Easily access creating/ destroying workspaces	
2	1	Designate workspace owners	
2.1	1	Workspace owners to have permissions to destroy workspace,	
3	1	Attaching workspaces to user profiles	
3.1	1	The ability to see all workspaces user belongs to on user profile	
4	2	Users can subscribe to Workspaces	
3	**1**	**Teams**	
1	1	Role based on permission for team	
2	1	Ability to assign users to various teams	
3	1	Assigned groups of tickets to various teams	
4	**1**	**Profile**	
1	1	Users can create a profile and log in	

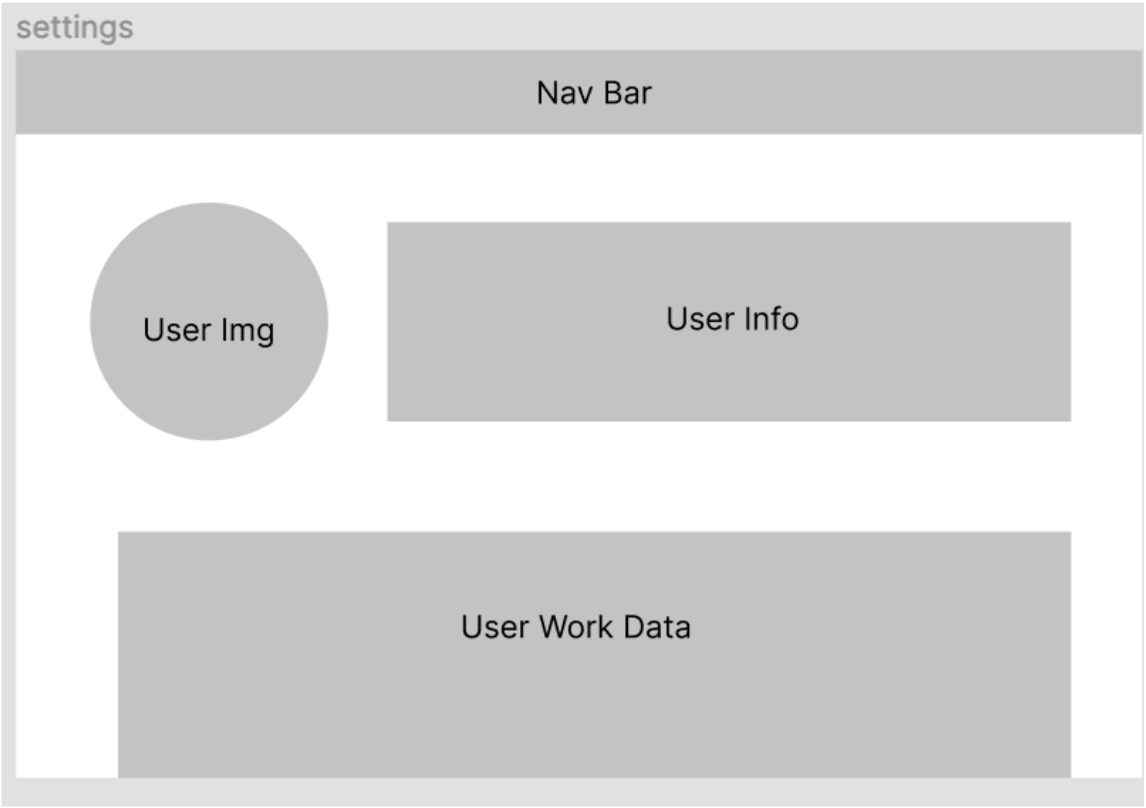
1.1	1	Assigned projects	When users are assigned a project, it will show up on their profile
1.2	2	Assigned tickets	When users ares assigned a ticket, it will show up on their profile
1.3	1	User name	Editable varying on permission given by organization
1.4	1	User title	Editable varying on permission given by organization
2	1	Basic session based authentication	
3	1	Users can change their password	
3.1	2	Password requirements	Organization will have ability to designate any password requirements (length, special characters, numbers, case sensitivity)
4	1	Users can view Workspaces and teams that they are assigned to	

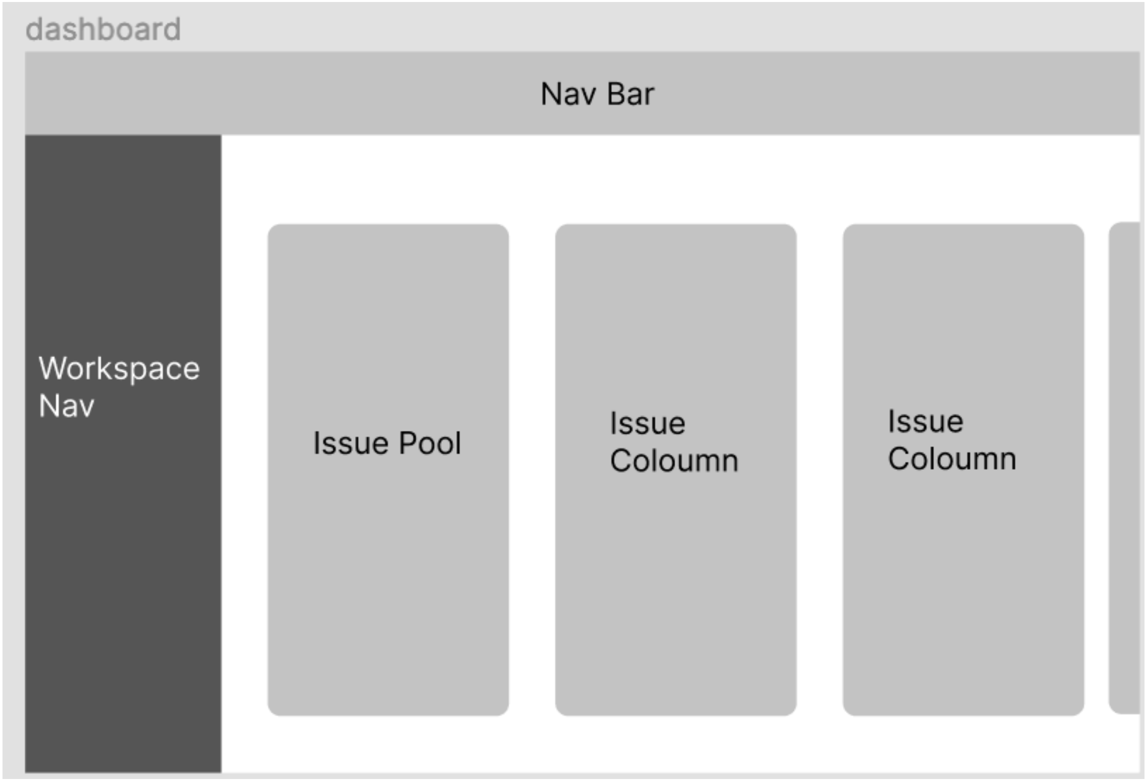
3. UI Mocks Ups and Storyboards

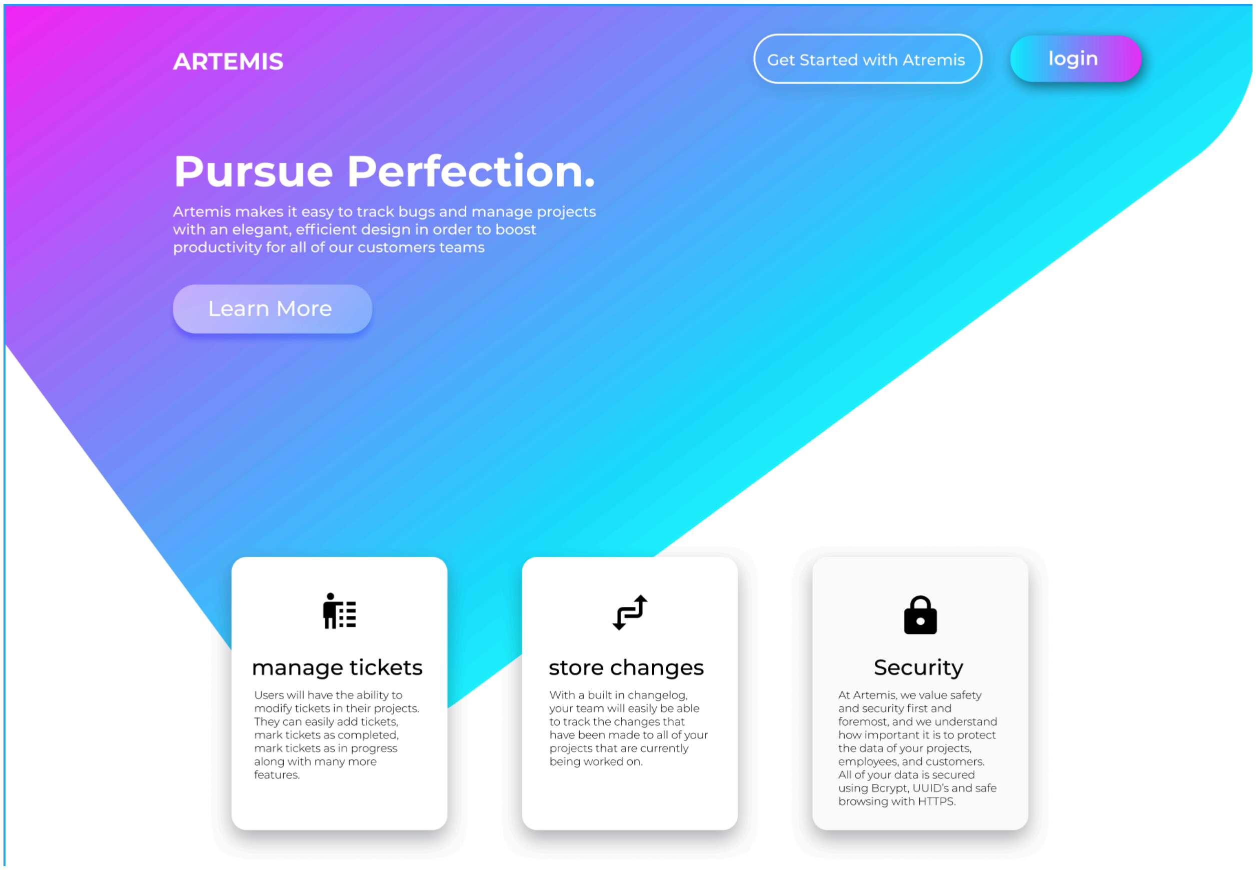



register
Nav Bar
Project Name
Logo
Registration Fields


log in
Nav Bar
Log In Fields
Project Name
Logo


settings
Nav Bar
User Img
User Info
User Work Data


dashboard
Nav Bar
Workspace Nav
Issue Pool
Issue Coloumn
Issue Coloumn



Pursue Perfection.

Artemis makes it easy to track bugs and manage projects with an elegant, efficient design in order to boost productivity for all of our customers teams

manage tickets

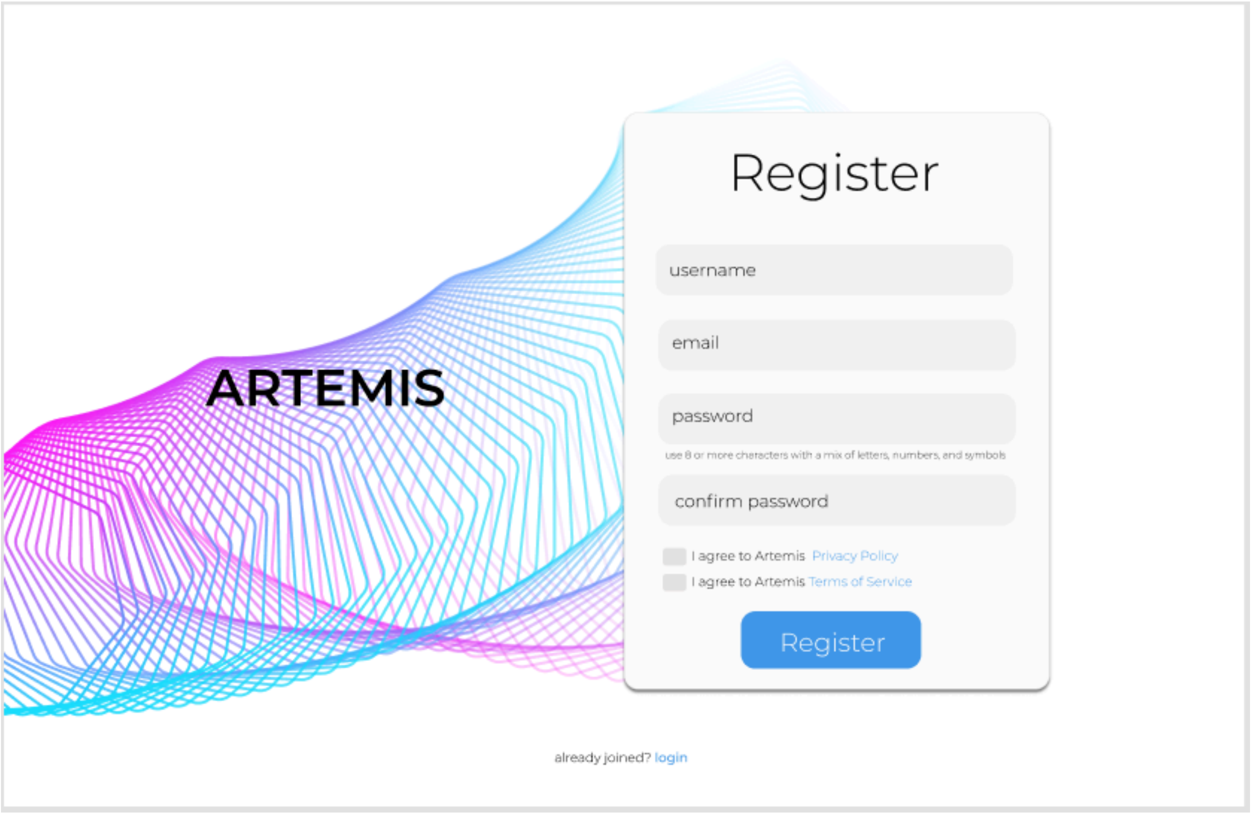
Users will have the ability to modify tickets in their projects. They can easily add tickets, mark tickets as completed, mark tickets as in progress along with many more features.

store changes

With a built in changelog, your team will easily be able to track the changes that have been made to all of your projects that are currently being worked on.

Security

At Artemis, we value safety and security first and foremost, and we understand how important it is to protect the data of your projects, employees, and customers. All of your data is secured using Bcrypt, UUID's and safe browsing with HTTPS.



Register

username

email

password

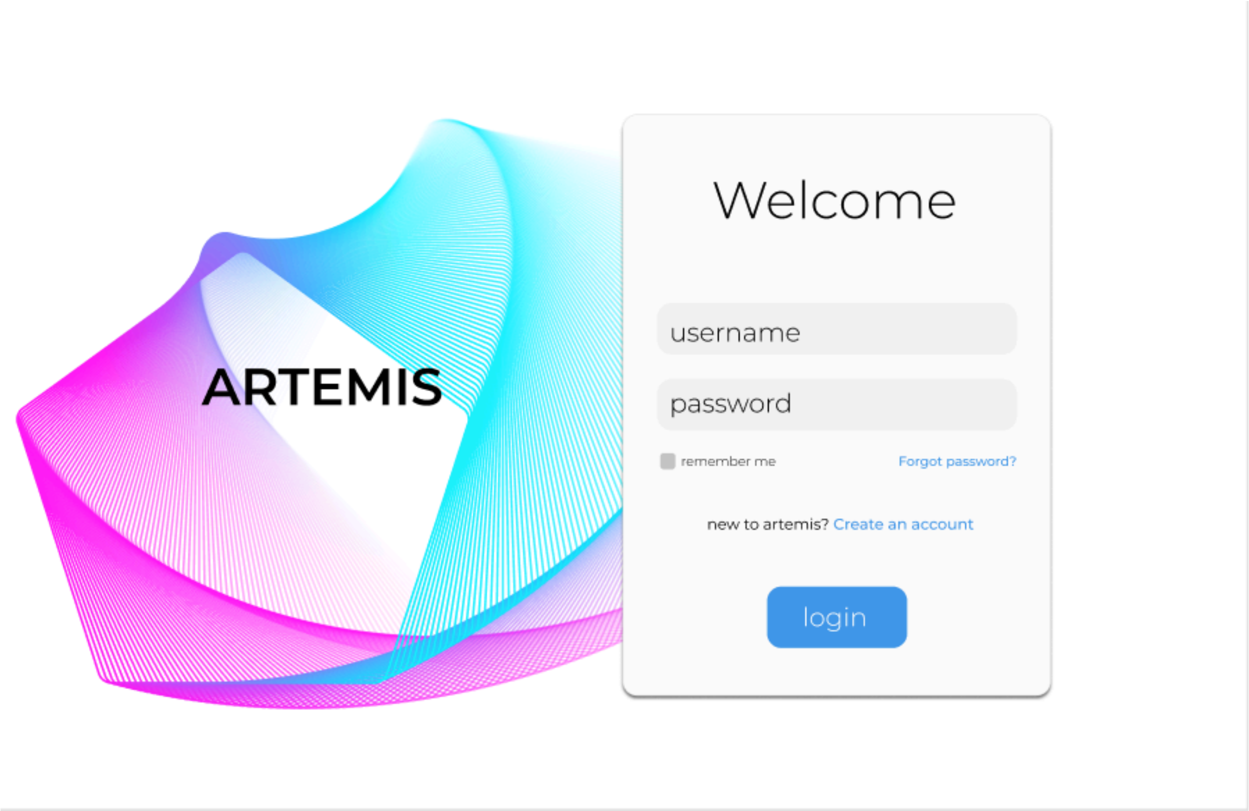
use 8 or more characters with a mix of letters, numbers, and symbols

confirm password

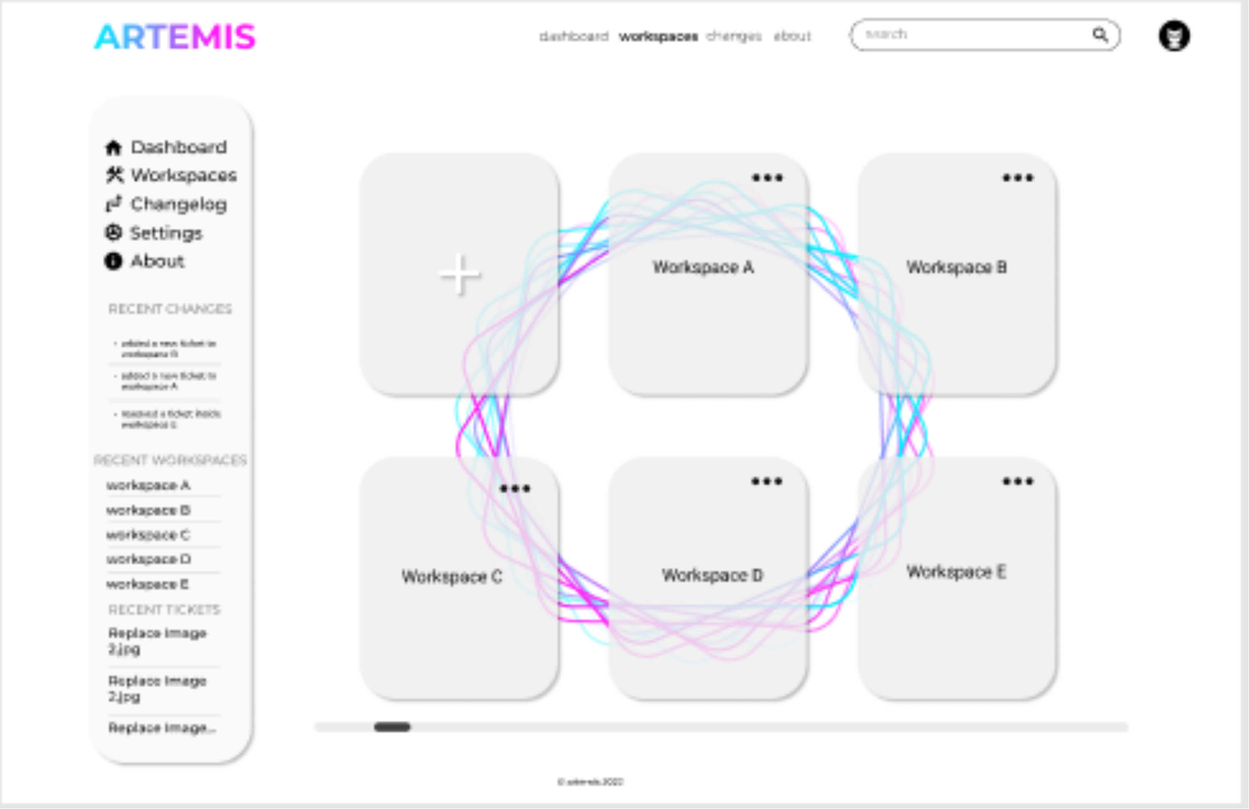
□ I agree to Artemis Privacy Policy

□ I agree to Artemis Terms of Service

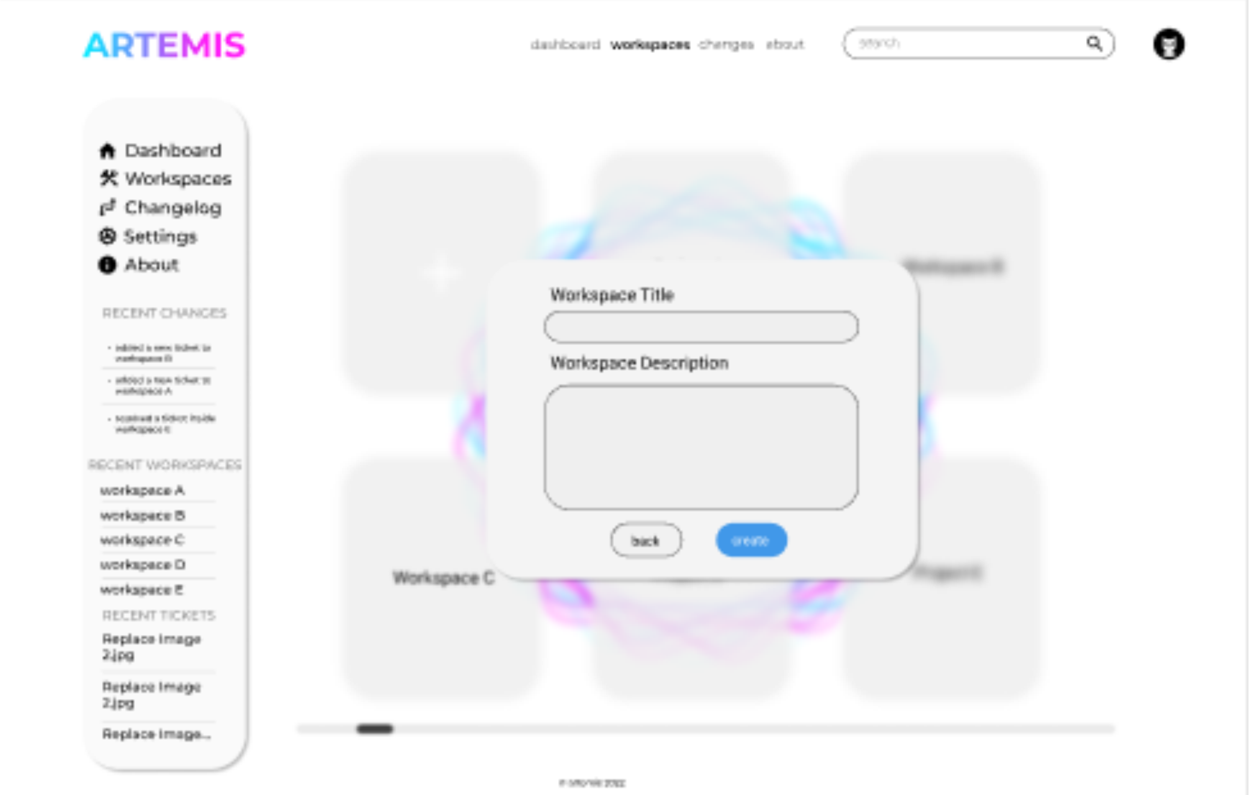
already joined? login

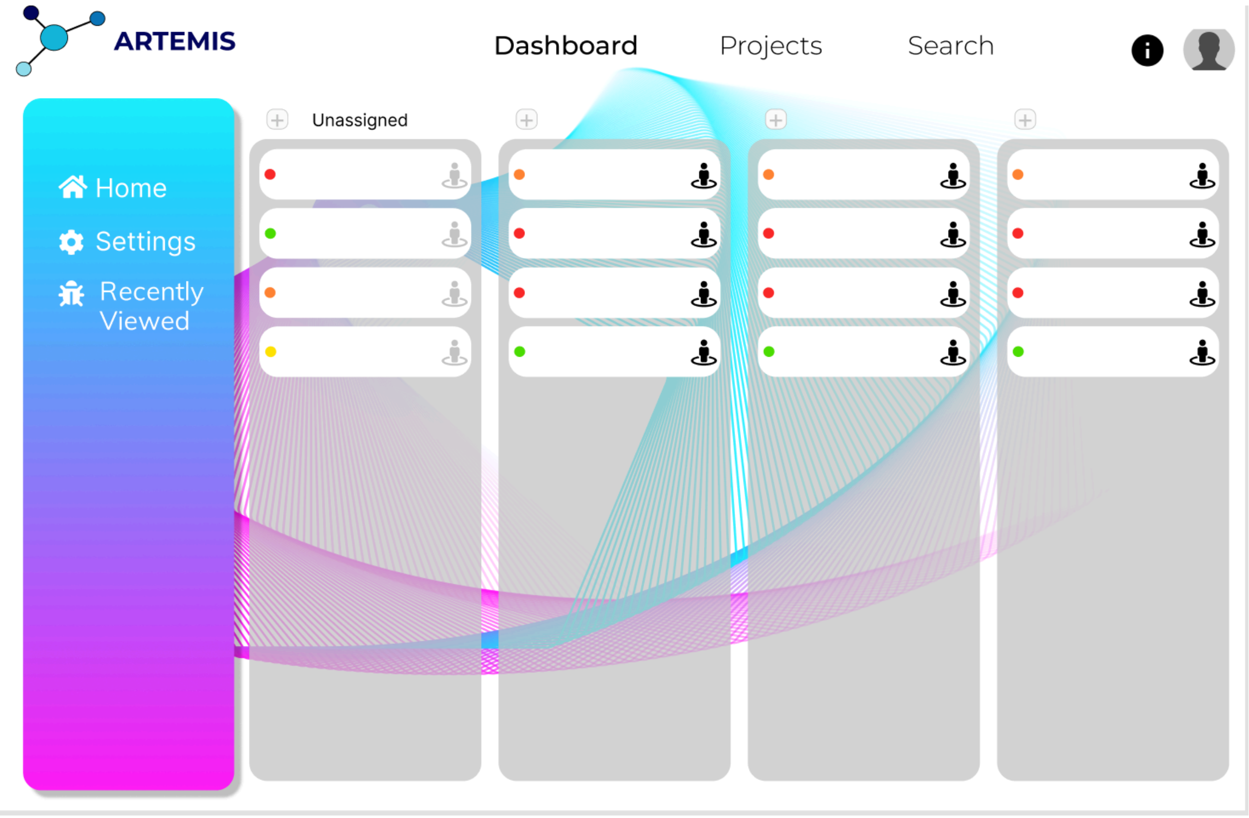


view projects page

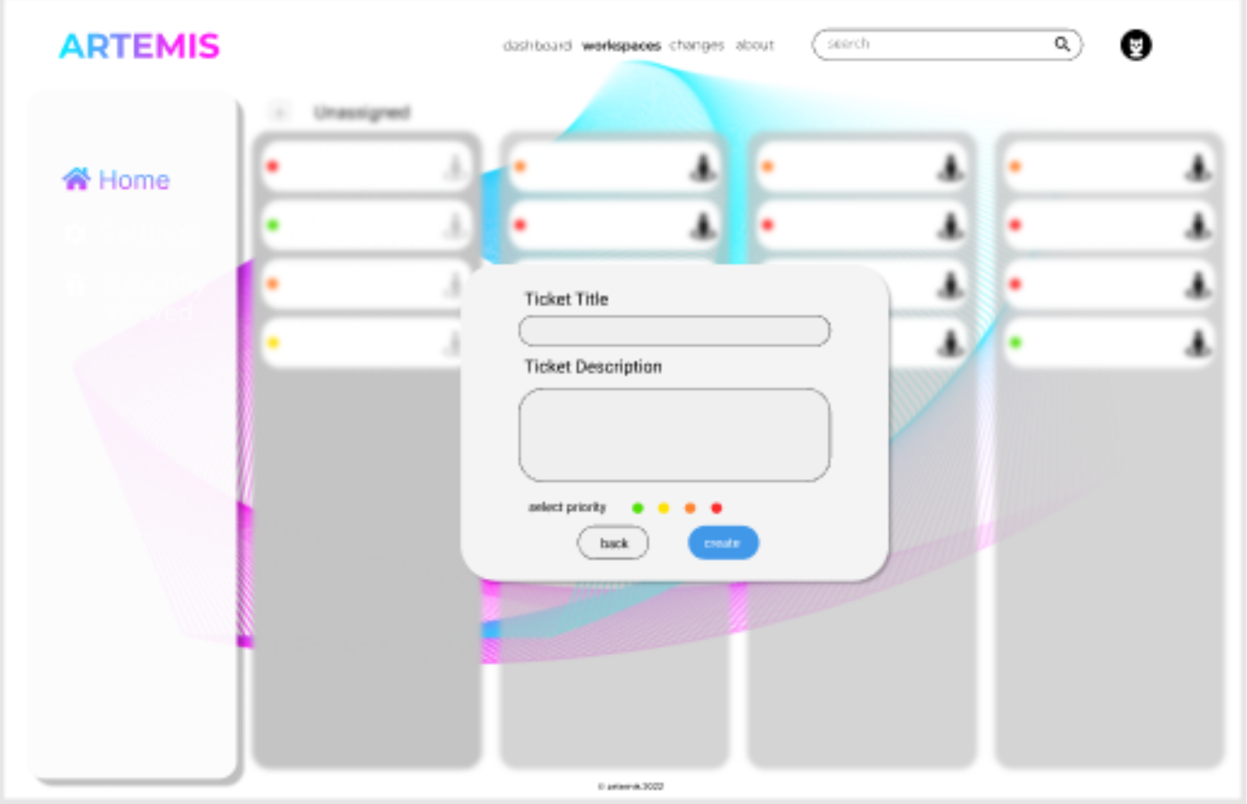


creating a new project





creating a ticket screen

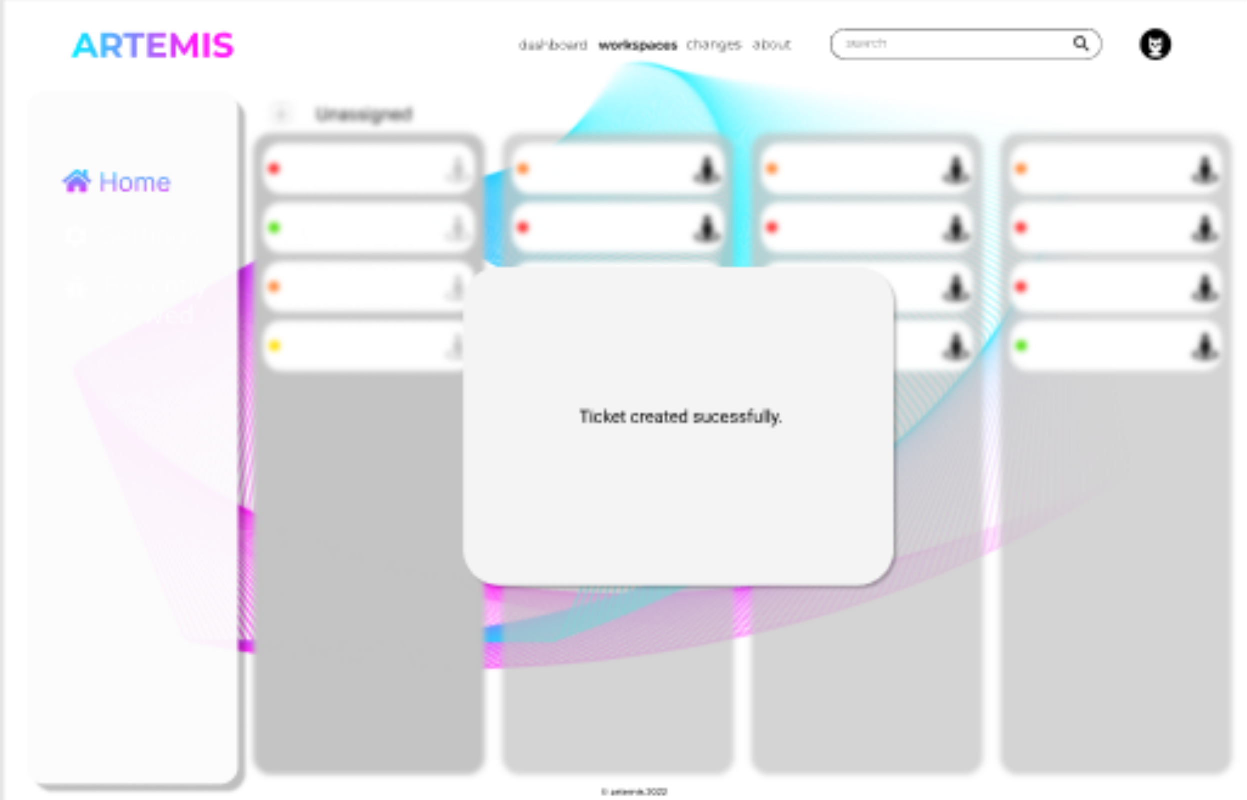


Ticket Title

Ticket Description

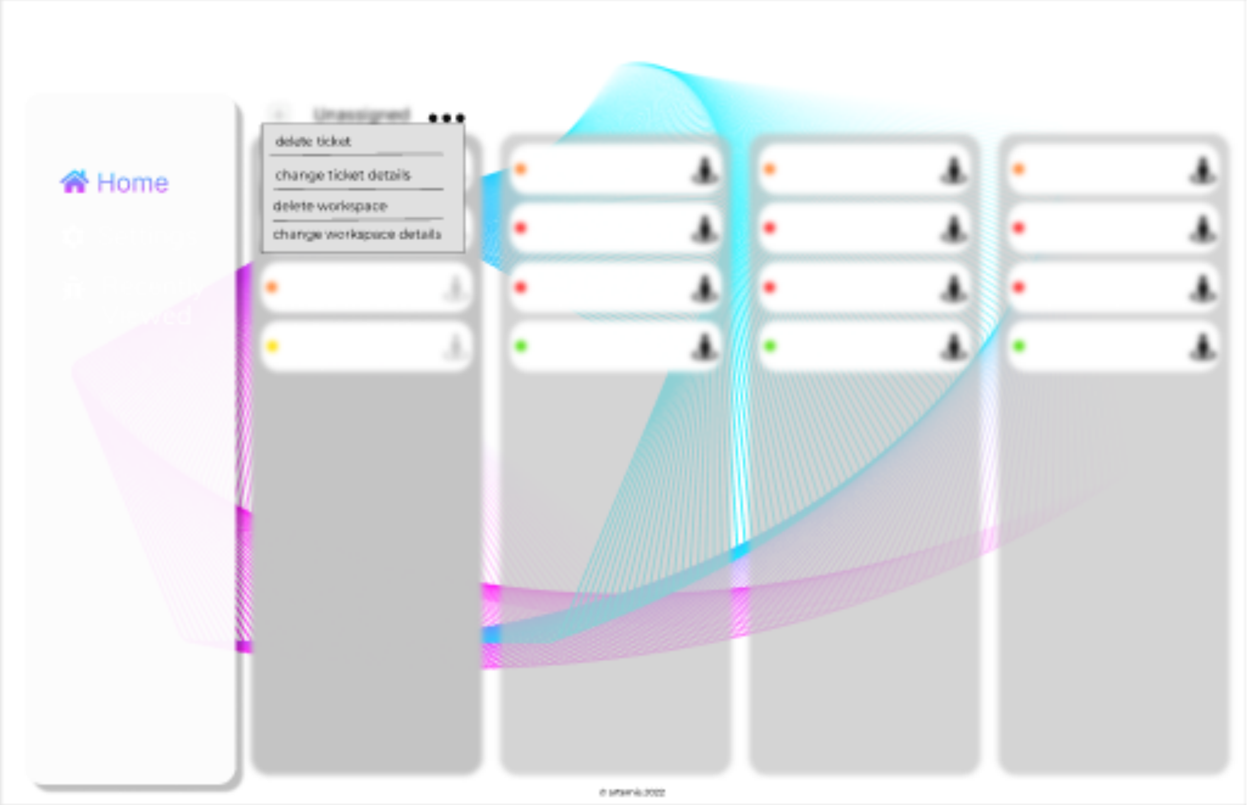
select priority ● ● ● ●

ticket confirmation screen

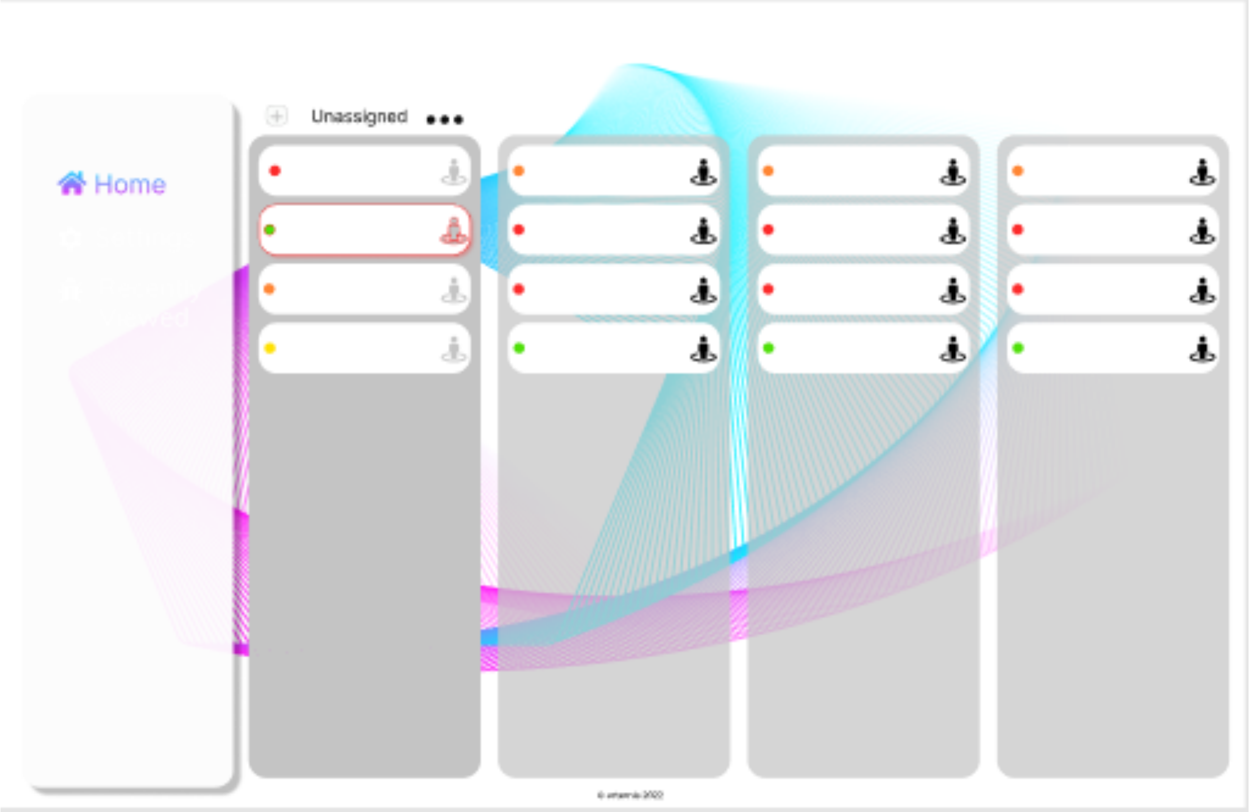


Ticket created sucessfully.

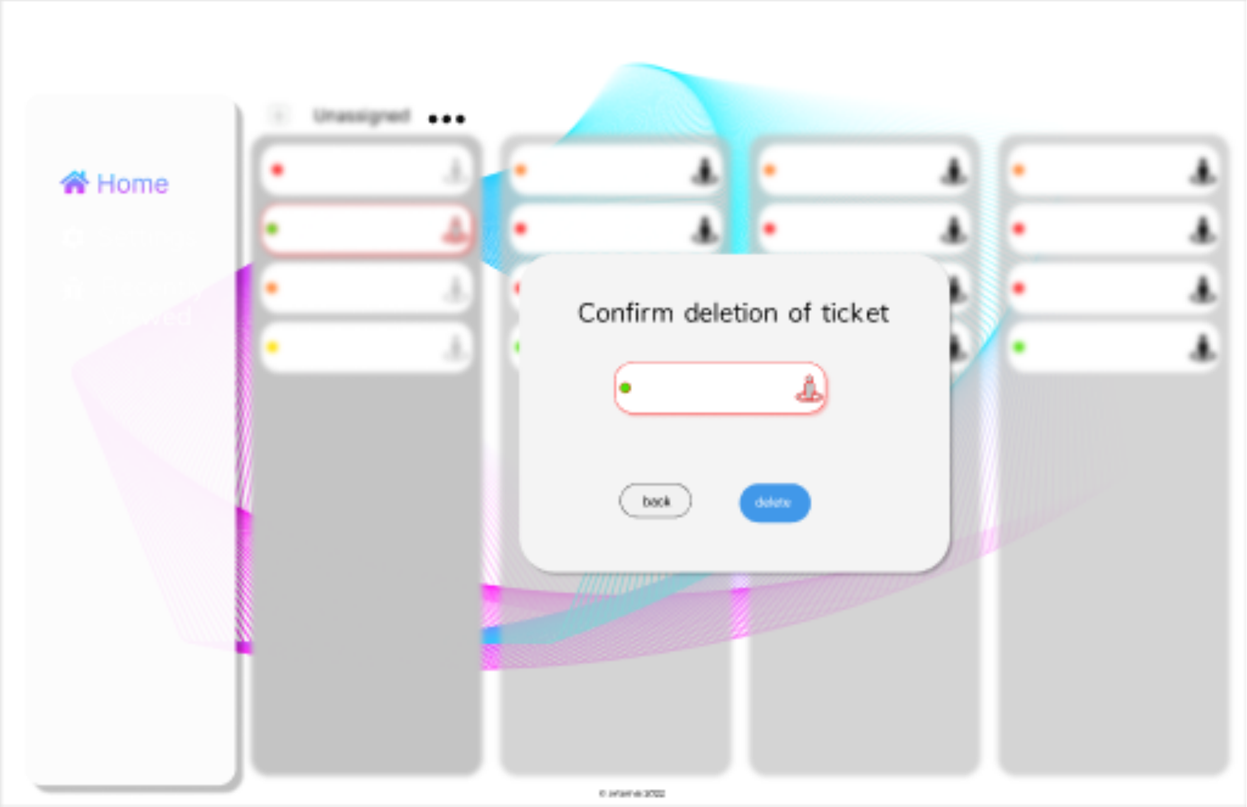
delete tickets screen



delete ticket highlight

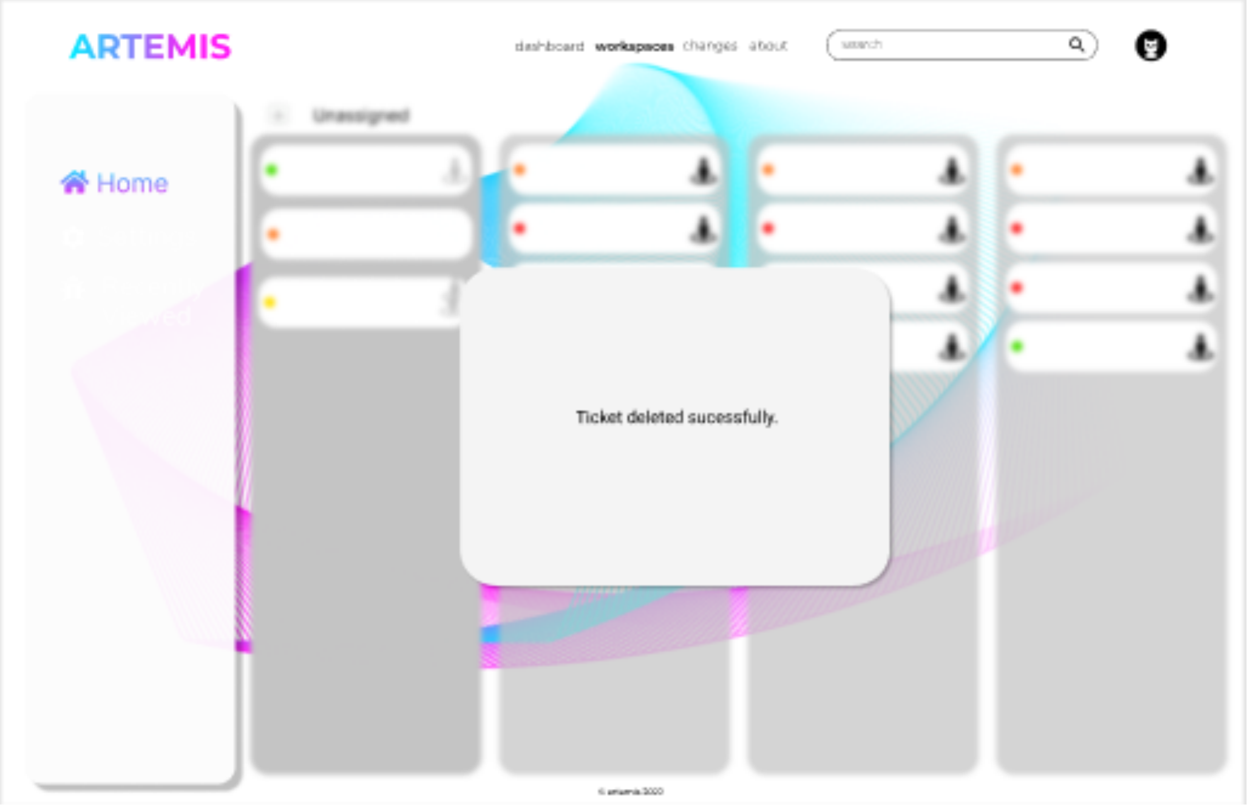


delete ticket confirmation

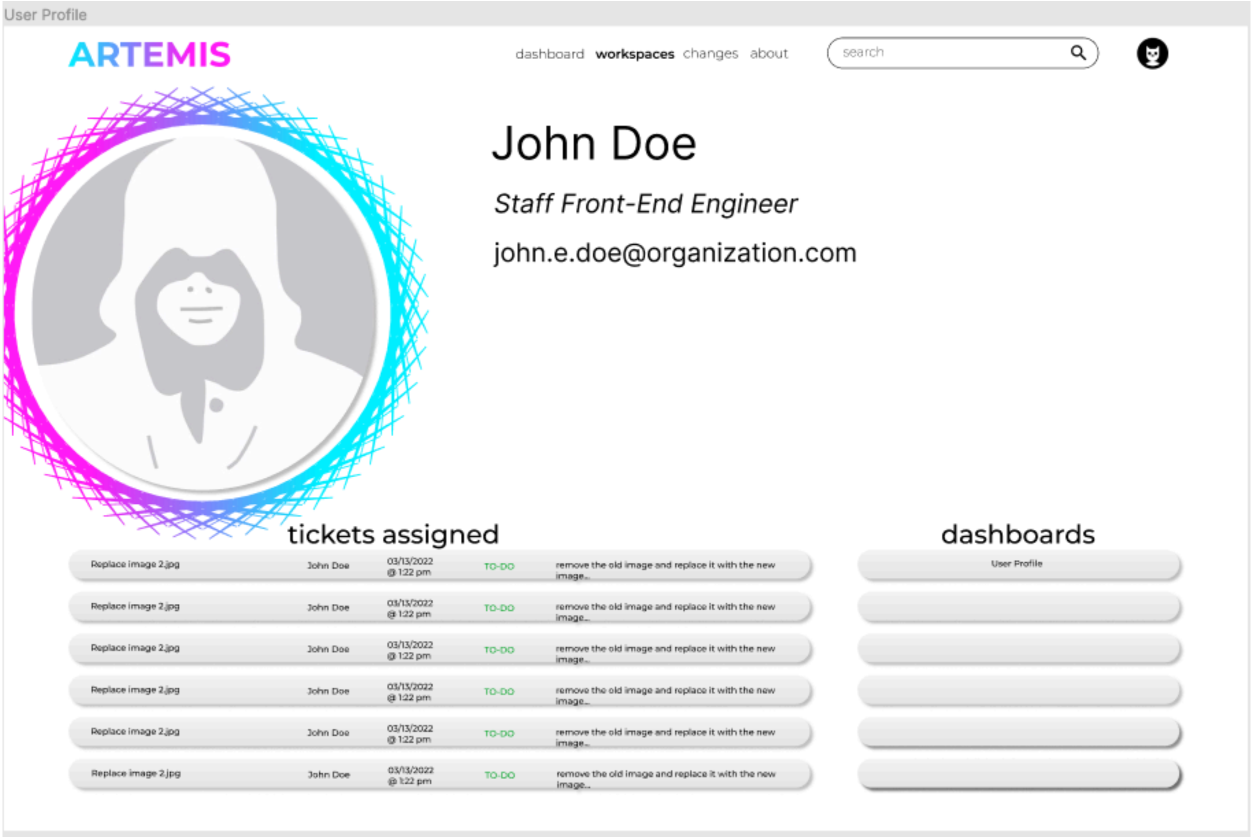


Confirm deletion of ticket

ticket confirmation screen



Ticket deleted sucessfully.



John Doe

Staff Front-End Engineer

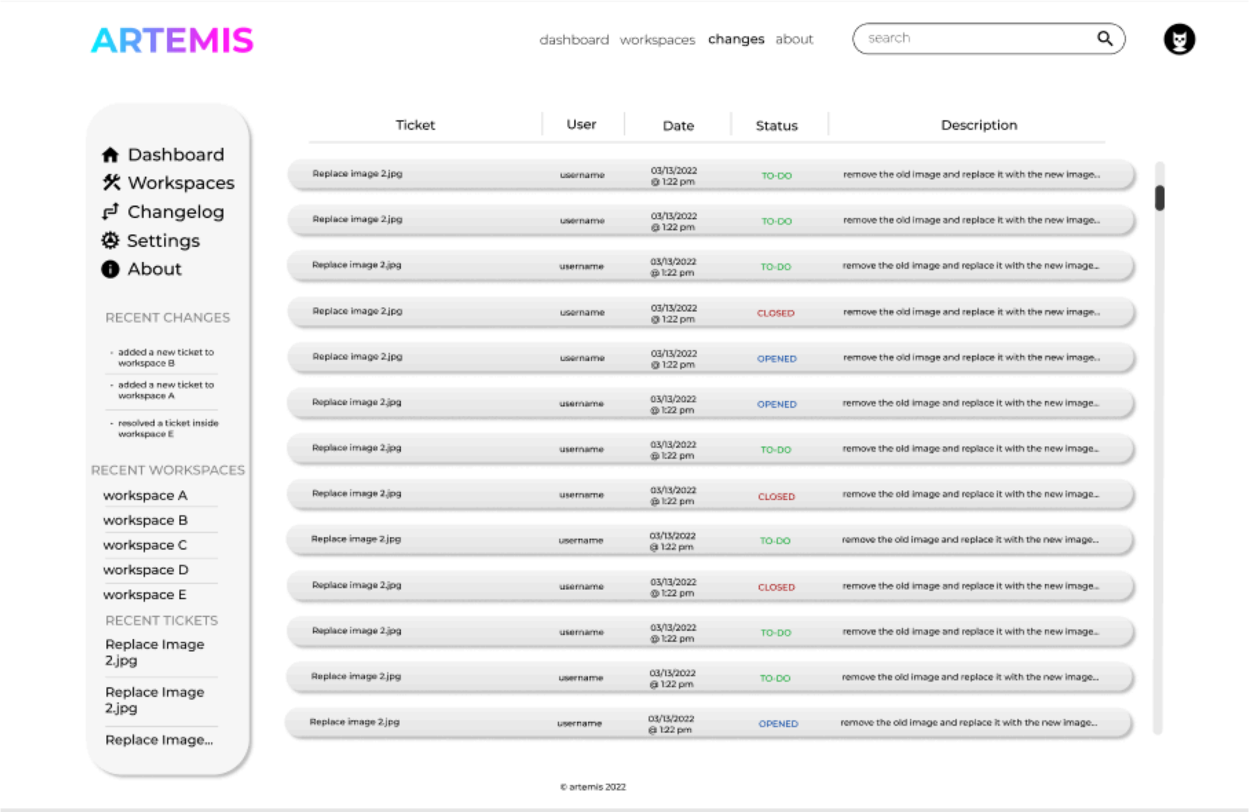
john.e.doe@organization.com

tickets assigned

Replace image 2.jpg	John Doe	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	John Doe	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	John Doe	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	John Doe	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	John Doe	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	John Doe	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...

dashboards

User Profile



RECENT CHANGES

- added a new ticket to workspace B
- added a new ticket to workspace A
- resolved a ticket inside workspace E

RECENT WORKSPACES

workspace A
workspace B
workspace C
workspace D
workspace E

RECENT TICKETS

Replace Image 2.jpg

Replace Image 2.jpg

Replace Image...

Ticket	User	Date	Status	Description
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	CLOSED	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	OPENED	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	OPENED	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	CLOSED	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	CLOSED	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	TO-DO	remove the old image and replace it with the new image...
Replace image 2.jpg	username	03/13/2022 @ 1:22 pm	OPENED	remove the old image and replace it with the new image...

© artemis 2022

UX Summary:
Through our discussions of the UI/UX, we realized we needed to work more on our flow. We decided the landing page would lead to the log in page—from there the user could choose to register. Once the log in form was filled out, the user would land on the workspaces page. The user then chooses the board they wish to observe. Tickets are available to create/edit/close out/ assign and delete. We kept the theme of the Blue/Pink mix for consistency; however, we tried to have it remain a small portion to not overwhelm the senses and be a clean, crisp site. We discovered we needed to change placements so the user could easily utilize the different tools available to them. After discussion of the adjustments, we made them in our GUI mock ups.

4. High Level Architecture, Database Organization

@Entity extends BaseEntity()
Organization
- Name (unique, non null)
- R: One to Many: Members (users)
- R: One to Many: Workspaces

Workspaces
- Name (unique, non null)
- R: Many to One: Organization cascade
- R: One to Many: Teams

Boards:
- Name (unique, non null)
- R: Many to One: Workspaces
- R: One to Many: Tickets

Teams:
- Name (unique, non null)
- Permissions? if time permits
- R: Mane to One: Workspaces cascade

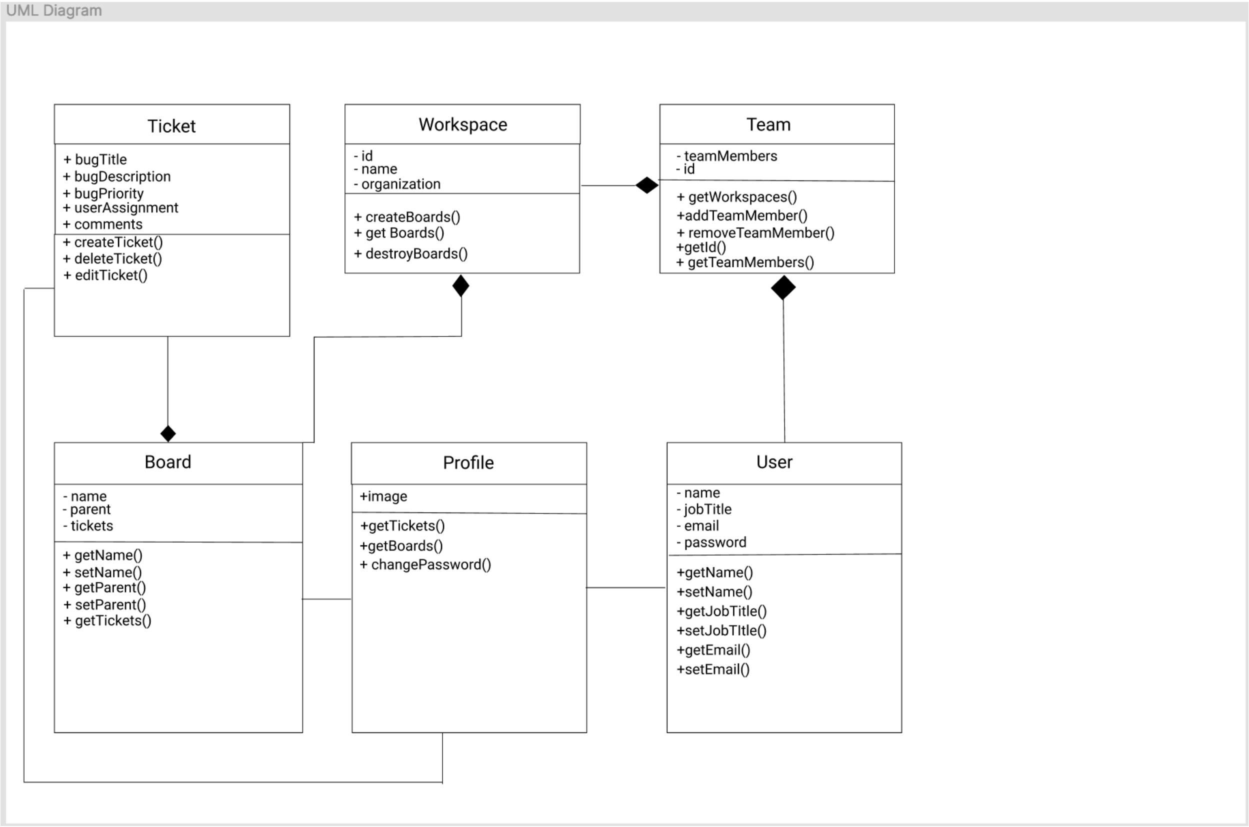
Tickets:
- ID (unique)
- Description (non null)

- Comments
- ClosedDate (nullable)
- OpenDate (nullable)
- R: Many to One: Board cascade

Users

- ID (unique)
- Username (unique, non null)
- Hashed Password (unique, non null)
- R: Many to One?: Organizations
- R: Many to Many?: Teams
- R: Many to Many: Boards

5. High Level UML Designs



6. **Key Risks**

   Skills Risks:

   Each team member has gone over technologies/languages they have not encountered before and continue to read up on them weekly.

   Schedule Risks:

   We have dedicated time in scrum meetings to dedicate to timelines and readiness for the upcoming milestones. We keep track of our responsibilities for each team member on Trello and utilize discord for off-the-cuff dialogue on any issues that have come up that need the hive mind to help figure out.

   Legal/Content Risks:

   All software utilized in building the project is being used with proper licensing and copyright.

7. **Project Management**

   The scrum meeting starts with a stand-up where each team member explicitly details the part of the project they are working on, their progress and any problems that have arisen. All members have a chance to discuss the problems along with any solutions they may be able to offer. Next is discussing the upcoming targets we are aiming to complete, breaking down to smaller units and assigning those to team members. Trello is being used for documentation and accountability. Discord is used at all other times to communicate with each other any and all information dissemination.

Vertical Prototype Home Test Page: www.thoughtgrove.com
Team GitHub: https://github.com/CSC-648-SFSU/csc648-spring22-04-team02

Software Engineering

CSC648/848

Milestone 1

Section 04 | Team 2

Artemis

2/28/2022

Brian Adams	badams2@mail.sfsu.edu	Team Lead
Joseph Kois	jkois@mail.sfsu.edu	Github Master
Brandon Cruz-Youll	bcruz5@mail.sfsu.edu	Scrum Master
Sherman Wong	swong42@mail.sfsu.edu	Backend Lead
Arslan Alimov	aalimov@mail.sfsu.edu	Backend Support
Jade Simien	jsimien@sfsu.edu	Frontend Lead

History Table:

2/14/2022	Everyone together as a team to create an executive summary, functional and non-functional requirements, personas, and competitive analysis.
02/28/2022	Fixed the M1 documentation, using the example M1 document as a guide to better adhere to guidelines provided

1.Executive Summary:

Artemis is a bug tracking application that will empower developers to track new and existing bugs. Being able to efficiently create and delegate responsibilities from a single dashboard enables development teams to prioritize and resolve issues, coordinate with quality assurance teams, and plan for sprints. Artemis condenses what the development team needs to know about issues and technical debt, in an easy-to-use application.

Artemis will provide a roles-based system that gives various users different privileges. A quality assurance team would be able to create tickets, a scrum master would be able to delegate issues, and developers would be able to mark tickets as being closed. This offers various teams the ability to work together, while reducing the possibility for error. Users will be able to see a changelog of the project history to understand what exactly has been changed with the program. Also, users will be able to view a roadmap of the current project they are assigned to, that way they can get an idea of what's to come.

Artemis strives to be a minimally designed, efficient, and effective project dashboard for tracking bugs. The ease of use throughout the system propels it above its competitors. The problem Artemis tries to solve is the issue of feature creep and bloat. Many other bug trackers will have features that are not necessary and oftentimes get in the way. Having a cluttered UI can make for a frustrating experience. Artemis strives to solve this with a minimalist design and easy-to-use features.

2. Personas:

Id: 01
Occupation: CEO of startup
Age: 40
Name: Sophia

About:

- She has been in the field for a while and has decided to take the plunge and start her own company.
- She has a Masters in Computer Science and did research in Machine Learning and Natural Language processing.
- She has worked for well-known companies such as Google and Apple, further developing her skills in the field.
- She is known for her contributions to the field–including her research.

Attitude:

- She is a go-getter but still has empathy for others around her.
- She helps lift people up, rather than see them as stepping stones to better places.
- She has always been somewhat of a risk-taker, but embarking on this startup adventure has led her to be more adventurous.
- She roots for the underdog and seeks to help other startups that she sees as great innovators.

Skills:

- Fluent in Python, R, JSON, Java, C, C++
- Expert in voice recognition and language modeling

Goals:

- Continue developing start up and work on getting VC funding

Sophia will use the application to organize her business. This will be a tool that will help maintain collaboration and communication across various teams. Sophia will not have to work with the application directly, rather she will be in charge of maintaining various teams that work with the application. Organizational productivity and fluid collaboration are the most important aspects of this application to Sophia.

Id: 02
Name: Noah
Occupation: Software Developer
Age:27

About:

- He came from another country to pursue his dream career in the United States.
- He is knowledgeable and has completed his education, such as Bachelors in Computer Science
- He is on his third job, however, he gained a lot of experience from his current job. He learned a lot about backend and front end development. He gained a lot of experience, product design knowledge and development skills from his past jobs.
- He loves challenges, teamwork, communication, and complicated tasks.

Attitude:

- Hard-Working, loves to learn new things especially in coding.
- Patient with people, especially his co-workers.
- Loves to listen and criticize work of his own and his teammates.
- Always positive and fun to interact with.

Skills:

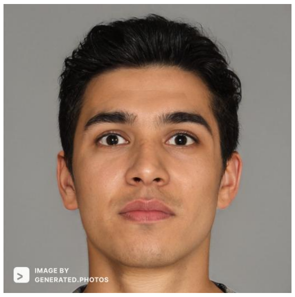
- Fast Typing
- Knowledge of languages such as: Java, Python, Javascript, JSON, C,C++, C#, Swift

Goals:

- Wants to become rich by 30 years old.

Noah will use the application as a developer. He will be assigned tickets that he will need to work through and close. The tickets Noah will most likely be assigned include bug fixes and refactoring. Noah will be able to keep notes of tickets in the application itself, and have the ability to share those notes with the rest of the workspace. Ease of use and the ability to organize his task queue in a logical way are the most important things for Noah.

Id: 03
Occupation: College Intern (Software Developer)
Name: William
Age: 22



About:

- Hard working student their whole life, very excited to get into the world of software development.
- Their first time working at a startup.
- They are in their last semester of college.
- Has never used software development tools before.
- They are interested in creating their own startup company someday.

Attitude:

- Frequently pulls all-nighters to debug code.
- Very sluggish in the morning, works best at night.
- Can be irritable sometimes when they forget to take breaks and eat.

Skills:

- Fluent in C++, C#, Java, Typescript
- Computer Networking
- Socializing

Goals:

- Create their own startup someday.
- Be the next Jeff Bezos.

William is a very driven student, who enjoys learning about new technology. He really appreciates a simple to use application that doesn't require reading lengthy tutorials to be productive. He is familiar with tools like Trello and finds drawing feature parodies between the two help him better understand how to use our application. William really values simple but powerful user interfaces.

3. User Stories:

ID	Persona	Wants (functionality or feature)	So that (benefit)	Constraints
1	Sophia	Ability to see bugs priority and easily view that they are assigned to someone	She can ensure her product remains useful	-The UI should be easily readable
2	Noah	Wants an easy to use ticket system to solve customers problems and needs. He also wants to be able to share the notes from tickets with the rest of team	He can fix bugs and update the website to customer needs. He wants the product to be always updated and	- Ticket system should be easy to use for everyone - The tickets must be written in a language that will be understandable to everyone.
3	William	Learn how to use software development tools in a professional setting.	Development and debugging can go at an organized and smooth pace.	-The tool should be intuitive for new users.

4. Data Definitions:

Name	Definition	Usage
Bug	Behavior that is unexpected or unintended	Tickets can cite known bugs to fix. Developers will be able to associate known bugs with tickets.
Ticket	A single task that needs to be done.	There will be lists of tickets that developers can be assigned to. These tickets can be closed once the task has been completed. Tickets can include tasks

		like bug fixes, maintenance and refactoring.
Close	The completion of a task, to mark a ticket (task) as being complete.	Developers will be able to close their tickets after the task has been completed.
Workspace	A collection of tickets, and potentially teams that belong to a particular organization. Workspaces contain all the information for a specific team or project, and are containers for tickets to be created and closed in.	Workspaces serve as a container to organize all the data for any given project or team.
Team	A collection of users who have specific permissions	Teams are used to organize users and what they can alter within any given workspace. If you belong to a developer team you can close tickets, if you are part of a QA team you can create Tickets etc.
Organization	An entity that can have multiple workspaces and teams.	Organizations provide a way for owners to maintain various workspaces.

5. Initial list of functional requirements:

Tickets:

1. Ability to create, and modify tickets
2. Users should be able to assign tickets
3. Being able to mark tickets as being closed
4. Ability to attach stack traces and descriptions to tickets
5. Previously closed tickets are viewable

Workspace:

1. Create, destroy workspaces
2. Designate workspace owners
3. Attaching workspaces to user profiles

4. Users can subscribe to work spaces

Teams:

1. Role based permission for teams
2. Ability to assign users to various teams
3. Assign groups of tickets to specific teams

Profile:

1. Users can create a profile and log in
2. Basic session based authentication
3. Users can change their password
4. Users can view organizations, and teams that they are assigned to

6. List of non-functional requirements:

Accessibility and Compliance:

1. The application is useable for those with vision impairments (correct contrast, predictable interface, readable text etc)
2. The application respects user privacy and does not unnecessarily track users behavior.

Security:

1. All sensitive user information is either encrypted or hashed
2. Application uses HTTPS
3. Use HTTP only and secure cookies

Performance:

1. Assets are small and lazy loaded
2. Use a bundler that tree shakes, minifies and transpiles code that is compatible with older browsers.

Design:

1. Create a UI that users can intuitively navigate
2. Use color to place emphasis on specific items of interest
3. Use readable text
4. Avoid bloating the UI with options that are not relevant to the current view

7. Competitive analysis: Jira + Trello

Jira +Trello	Artemis

Roadmap feature to organize the duration of development	Simple interface that is designed with productivity in mind
Agile reports that track time management, created issues, etc.	Tracks issues, and the state of the issues through development
Integrates a team’s commonly used apps into the workflow	Stand-alone application that does not require any third-party integration
Checklists for sub-tasks	Issues can be prioritized and be in various states
Can leave comments for the team or private notes on cards	Devs can comment on the bugs, and list blockers

Jira and Trello approach project management with superfluous designs and features that can be overwhelming. Artemis is planned to have a simple interface and result in a more intuitive workflow. While Trello allows checklists for sub-tasks, Artemis organizes tasks/issues based on priority levels. This encourages productivity by allowing teams to focus on the most pressing tasks to be completed rather than overwhelming the flow with a plethora of low leveled tasks to be checked off.

8. High-level system requirements:

** No. refers to the item number & priority of item*

No.	**Requirement**	**Description**
01	View Ticket	Able to view tickets that were created
02	Create Ticket	Create tickets that need to be attended to
03	Roadmap	View a roadmap of the current project or future projects
04	Changelog	Keep track of all changes made to the codebase
05	Resolved issues	Be able to see previously completed tickets
06	Username & password	Give users the ability to sign up/ log-in, in a secure manner
07	HTTPS	Be able to support HTTPS for secure browsing/use
08	Design	Minimal design to promote efficiency and usability

09	Maintainability	All components structured in a way so that maintainability is easy

9. High Level frameworks and tools

Backend	Express TypeScript/JS TypeORM or Sequelize MySQL Session Bcrypt UUID
Frontend	TypeScript NextJS or React CSS Modules SWR Zustand Framer Motion UUID
Design	Figma
Version Control	Git GitHub
Testing	Postman (End to End) Jest (Unit Test) Cyprus (End to End)

10. Team:

Brian Adams	badams2@mail.sfsu.edu	Team Lead
Joseph Kois	jkois@mail.sfsu.edu	Github Master
Brandon Cruz-Youll	bcruz5@mail.sfsu.edu	Scrum Master
Sherman Wong	swong42@mail.sfsu.edu	Backend Lead
Arslan Alimov	aalimov@mail.sfsu.edu	Backend Support
Jade Simien	jsimien@sfsu.edu	Frontend Lead

11 .Checklist:

- Team found a time slot to meet outside of the class Done
- Scrum Master shares meeting minutes with everyone after each meeting. Done
- Github master chosen Done
- Everyone sets up their local development environment from the team's git repo. Done
- Team decided and agreed together on using the listed SW tools and deployment server Done
- Team ready and able to use the chosen back/front-end frameworks. Done
    - For each technology (front/back-end/DB/cloud), the team decides who will lead the study of each technology and what will be output of the (feasibility) study within one month. On Track
- Team lead ensured that all team members read the final M1 and agree/understand it before submission Done