Team 1

Team 1 - Matrix

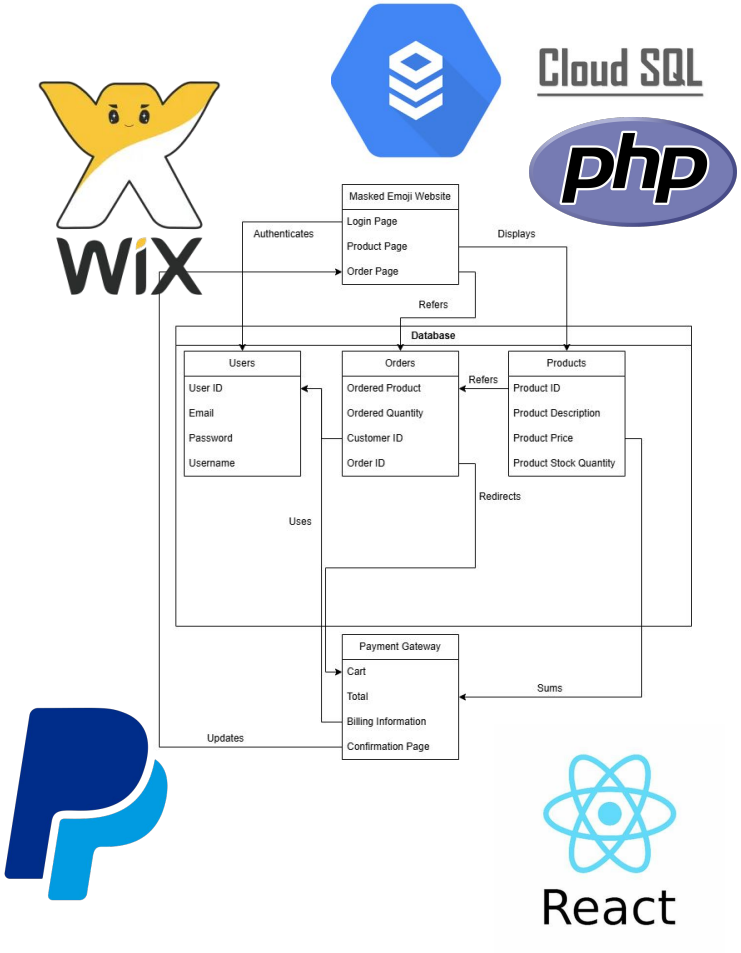
Matthew Alvarez, Ayomide Ajibola, Luis Nunez Gonzalez, Harsh Hirpara, Edmund Owusu-Ansah, Ajendy Rodriguez, Jordan Ruffin

Project Overview

- **Masked Emoji Website**
  - Create an ordering system for the clothing brand
  - Provide users with news and information about the brand
  - Give client a solution to the problem of having to rely on merchant apps to have a storefront presence
- **Client and the Customers**
  - Need to streamline customization and checkout
  - Need for a centralized outlet for both community interactions and products
  - Need for a system that logs orders in case of disputes.

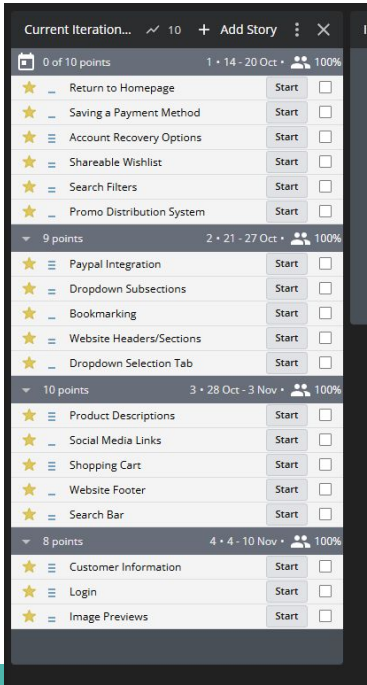
Requirements Overview

- **Functional Requirements**
  - Login page
  - Order page
  - Database access
  - Payment gateway
- **Non-Functional Requirements**
  - Payment security
  - Database Performance
  - Website Hosting



Product Backlog and Sprint Planning

- **Highlights of Product Backlog**
  - The backlog is split up into 4 sprints consisting of 3-6 user stories in each sprint. Each sprint has a velocity between 8-10 points.
  - Key User Stories:
    - Paypal Integration
    - Shopping Cart
    - Customer Information
  - Priority of the key user stories:
    - High Priority because they will take the longest time to complete.
- **Sprint Planning Strategy**
  - How are sprints planned? Pivotal Tracker, Meetings
  - Which user stories should be selected? Highest Priority, Business Needs
  - How is progress measured? Sprint Velocity



Lessons Learned So Far

- **Customer Interaction**
  - Meetings with the client are weekly.
  - Thinking like the customer, we focused on translating the clients needs into system requirements. For instance, prioritizing a user-friendly checkout system which includes the integration of PayPal for seamless transactions.
  - Client was very heavy on customer interaction so we plan on implementing the newsletter

- **Team Member Interaction**
  - Team Meetings are held both online and in person.
  - Experiences regarding working together has taught us to overcome challenges by supporting each other, making the project smoother and more rewarding.

- **Non Technical Aspects**
  - Work Life Balance is manageable by setting clear meeting schedules and spreading out tasks evenly has helped us maintain a healthy work-life balance.
  - With time management, it has taught us that breaking tasks into individual sprints helps us meet deadlines and keep our project on track.

Team 2

Green Lane Development

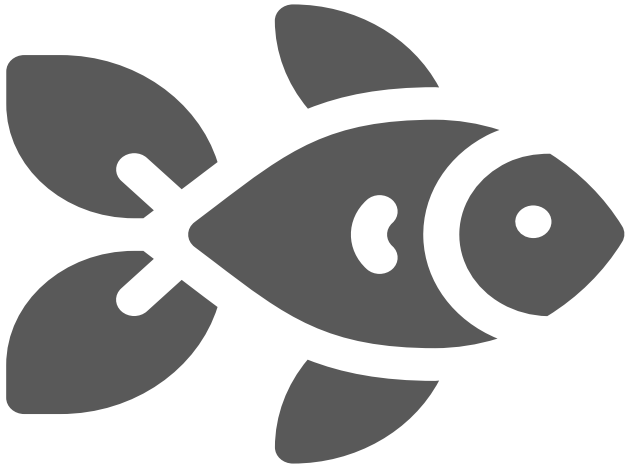
Team #2: Alexandra Mallqui, Basil Mathai, Brodie Berger, Christopher Vallas, Joshua Figueroa, Fernando Marques

Problem

- Hawk Island Marina in Delanco, NJ
- Family-owned by John & Karen Berger
- Primarily paper-based operations, limited online presence

Goal: Expand online presence for customer accessibility and operational ease

How will we solve the problem?



- Meeting User Needs
- Delivering Functional Solutions
- Ensuring High System Performance
- Providing Real-Time Data Insights
- Maintaining Strong Security and Access Control

Here are the tools we are using:

1. cPanel®
   Web server hosting control panel
2. Visual Studio Code
   Our choice of IDE for development.
3. phpMyAdmin
   Backend management with PHP and MySQL

Rundown of sprint(s):

1. Backend Core Development
   - Setting up backend infrastructure. Create the database structure for customers, marina spaces, reservations, incidents, tasks, etc.
   - Integrating weather API's
   - Set up security to secure storage of sensitive data and user authentication
2. Staff Operations
   - Develop staff dashboard for managing operations (managing customer records, boat assignments, incident logging)
   - Task management (implement features to create, assign, and track staff tasks and maintenance operations)
3. Customer Oriented Features
   - Develop customer interface(viewing marina spaces, making reservations, accessing tide/weather information)
   - Implement notifications for reservations and renewals
   - Test end to end flow
4. Manager and Reporting Features
   - Management dashboard (revenue reports, customer analytics, pricing updates)
   - Any refinements

Feedback

Most of the features of the web application were very well received. Certain changes were suggested to make it more in line with how a boating marina would actually operate.

For example, event planning is done by outside companies and organizations.

Boating marinas do not offer checkups or inspections, this aspect had to be removed as well.

Client Turnover

The turnover can be done by transferring the website files to the current web server paid for by our client.

Comprehensive documentation must be made and distributed to the staff so that the turnover is as clean as possible. Staff would likely need to be trained as specified by the documentation. Client use would likely be simple to understand.

Completion Time

Our project is projected to be completed on time for December.

Team 3

Team 3, Sinister 6

By: Allan Bardales, James Ford, Daniel Newlove, Jaynil Patel, Angelo Perez, Darell Samedy

Project Overview

01

The Problem

The main problem our client has involves no advertisement for their non-profit, limited donation accessibility, and minimal security of all members and donations. The project will provide both interactive system that allows users to sign up and be potential members along with a payment system where users can donate to the non-profit.

02

Goal

The main goal is to create a interactive system for our client's nonprofit organization and allow the website to generate potential members by having users sign up and having their information stored within a database, along with their donation information and implementation of security.

03

Project Users and Needs

Members :
Needs

- Easy sign up process
- Ability to donate through payment service

Admins
Needs

- View all active members currently signed up.
- Access to member's database

Product Backlog and Sprint Planning

Backlog
1. User Registration 2. Member Login: 3. Make Donations: 4. View Donation History: 5. Making a Joint Venture: 6. Home Page: 7. Access Donation Records Dashboard 8. View All Donation Records 9. Filter Donations by Date Range 10. Manage Database

Sprints
Sprint 1: 1, 2, 5, 6,
Sprint 2: 3, 4, 10
Sprint 3: 7,8,9

Functional and Nonfunctional Requirements.

Main Functional Requirements:

- User Login
- Donation System
- File Storage
- Homepage Information

Non-Functional Requirements:

- Performance
- Security
- Scalability
- Reliability
- Usability
- Maintainability
- Compatibility

Tools & Resources

Powered by Cloud Services

We want to design a system that will permanently store organization details, member services, all important and host the system.

Reasons Why!

- Cost Efficiency
- Scalability
- Data Security for Members and Organization
- Disaster Recovery
- Uptime
- Analytics

Azure Virtual Machines
Azure Storage & DB

What We Have Learned So Far

What We Learned

We learned to tailor our design to the client’s needs.

To balance internal deadlines in regards to our client requests, in order to efficiently utilize our time.

That our user stories allowed us to explain complex features in simple, client-friendly terms, ensuring they understood the system’s functionality.

What We Aim to Achieve

We aim to adjust our design approach to meet our client's evolving needs.

To deliver a system that’s not only functional but also client-friendly, allowing the client to manage it independently.

To ensure the system’s design is both user-focused and aligned with the client’s mission.

Team 4

Team 4
Cyber Zamurai
Cloud-Based Admin Scheduling Tool

Team Members
Leandro Perez, Cristian De La Rosa, Umar Abid, Stalin Zumba, Keanu Lee, Juan Patino

Solution Overview

- The Admin Scheduling Tool
  - An online course scheduling system designed specifically for Kean University. This tool streamlines the scheduling process for faculty, enabling efficient management of faculty members, assigned courses, and sections for each semester.
- Key Functionality
  - CRUD Operations: easily create, read, update, and delete records for faculty, courses, and sections
  - Rooms and Faculty: Simplify the allocation of classrooms and faculty assignments, ensuring optimal use of classrooms
  - User Friendly Interface: The interface is easy to use and connected to Keanwise to ensure the system is intuitive and accessible for all users.
- Simplification Process
  - The tool significantly reduces administrative burdens, allowing for faster and more efficient scheduling. The Cloud-Based Storage Integration ensures that data is securely stored and easily accessible.

Tools and Technologies

- HTML, CSS, JavaScript, Python(Programming Languages): languages for building the structure, styling, and interactivity of the platform.

Backend Technologies:

- Node.js & Express (JavaScript Backend Framework): Provides a scalable environment for handling server-side logic and APIs.
- Django (Python Web Framework): A high-level Python framework that simplifies back-end development with built-in admin functionality, security features, and a scalable architecture.

Other Technologies:

- Cloud-Based Databases (Azure SQL): Provides secure, scalable storage for faculty and course data, aligned with cloud hosting needs.
- Microsoft Azure (Cloud Platform): A cloud platform selected for hosting the application, offering scalability, easy deployment, and seamless integration with Microsoft 365.

Sprint Planning and Product Backlog

Backlog Management:

User stories were prioritized based on client feedback and critical system needs. The most essential functionalities, such as CRUD operations and real-time updates, were tackled in early sprints to ensure the core system was functional from the start.

Each sprint focused on delivering key system features that align with the client's needs:

- **Sprint 1**: Implemented CRUD operations for room and faculty scheduling, ensuring that basic data management could be performed.
- **Sprint 2:** Introduced cloud-based real-time updates to keep the scheduling system synced across all users.
- **Sprint 3:** Focused on refining the user interface and enforcing admin-only access to improve usability and security.

Each sprint was designed to last 3 weeks maximum in order to meet the final December deadline.

Client Feedback & Progress

- Client Feedback:
  - The client is very lenient when it comes how complex he want the scheduling system to be
  - The client was emphasizing that the importance of this project is for possible admins to be able access the database when needed
- Feedback Results:
  - As a group we were able to incorporate more user stories into our project and focus on needing to focus on the needs of potential admins that plan to view the database being created
- Client Turnover:
  - The client turnover will involve a seamless transition, allowing the client to easily access the file being presented with an explanation of what we incorporated into the project to ensure clarity.
- Project Deadline:
  - The project is estimated to be finished on time with no issues by December.

Lessons Learned

01 Collaboration- Working together with others to produce or create something.

02 Communication- Exchanging information about the project that can prevent delays and coordinating with team members and the customer.

03 Documentation- A record that provides information about the product.

04 Conflict Resolution- Coming to a common agreement on a dispute where all parties that are in involved and are satisfied with the outcome.

05 Delegating Responsibilities- Defining roles and assigning responsibilities to individuals on what needs to be done.

06 Scheduling- arranging or finding a time that works best for all parties involved.

Team 5

Team 5: Digital Dinosaurs

CopperShore Platform

Christian Franco, Samantha Gonzalez, Jolina John,
Peter Sweeney, Travis Matos, Isaack Traore

01.

Project Overview

Description:

- The CopperShore platform offers secure Google authentication, content creation tools, robust search and tagging options, user autonomy for post management, and customization features, all within an interactive environment managed by the site owner.

User Goals:

- Facilitating Collaboration
- Enhancing User Engagement
- Streamlining Content Discovery
- Ensuring Usability

User Needs:

- Secure Authentication
- Content Creation Tools
- Feedback Mechanisms
- Customizability
- Content Discovery

Requirements

Key Functional Requirements

Manage Posts

- This function gives the administrator the ability to manage the blog. Giving him the option to view, edit, delete, boosting and archive posts. They are also able to schedule their posting allowing them to upload their content at their desired time

Google Authentication

- This allows users to log in through google for a quick sign in experience. This requirement will be achieved through the use of OAuth 2.0 Google credentials.

Create Posts & Share Posts

- Users are able to create new posts, including images and descriptions of 3D printing projects. User will also be able to share posts with other users.

Nonfunctional Requirements

Website Loading Speed

- Website loading speeds are crucial because they enable users to quickly access the content they need. It'll be met through website hosting where we will be using Kean University server.

Traffic Control

- This function is crucial because it allows multiple users to post their desired content at any time, regardless of how much traffic the website is experiencing. We will be able to met this requirement by making sure our code is efficient reducing server usage.

Minimalistic User Interface

- A minimalistic GUI will enables users to quickly find specific content on the blog. We will meet this requirement by reducing clutter such as limiting elements.

Debugging Tools

- This function will detect, flag and handle errors. We will met this function by using applications such as visual studio code eliminate errors.

Content Moderation

- This function allows for content to be removed and monitored. We will meet this requirement by allowing users to flag and report inappropriate content.

Product Backlog + Sprints

Key User Stories

- As a user, I want to find 3D modeling projects easily using tags and search functions.
- As an owner, I want to moderate blog posts and comments to keep the platform focused and relevant.
- As a user, I want to sign in with my Google account to ensure a secure and authenticated experience.

Priorities

- Enhance searchability and tag integration for project discoverability
- Creation and deletion abilities for moderators to ensure appropriate content and comments are being uploaded
- Implementing a Google Authentication login for security and data tracking

Sprints

Sprint 1
- Obtain Google API Key, create front end login integration, and back end authentication
- Set up database and relevant tables
- Test Login

Sprint 2
- Set up automatic commenting and post updates to database tables
- Develop front end for posting and commenting functions
- Test functionality, account verification, and post verification

Sprint 3
- Create post filtering function for users
- Add post deletion, edit, and post boosting functions for moderator and connect this to database tables
- Test all blog functionalities for users, moderator, and owner.

Lessons Learned

Customer Interaction:

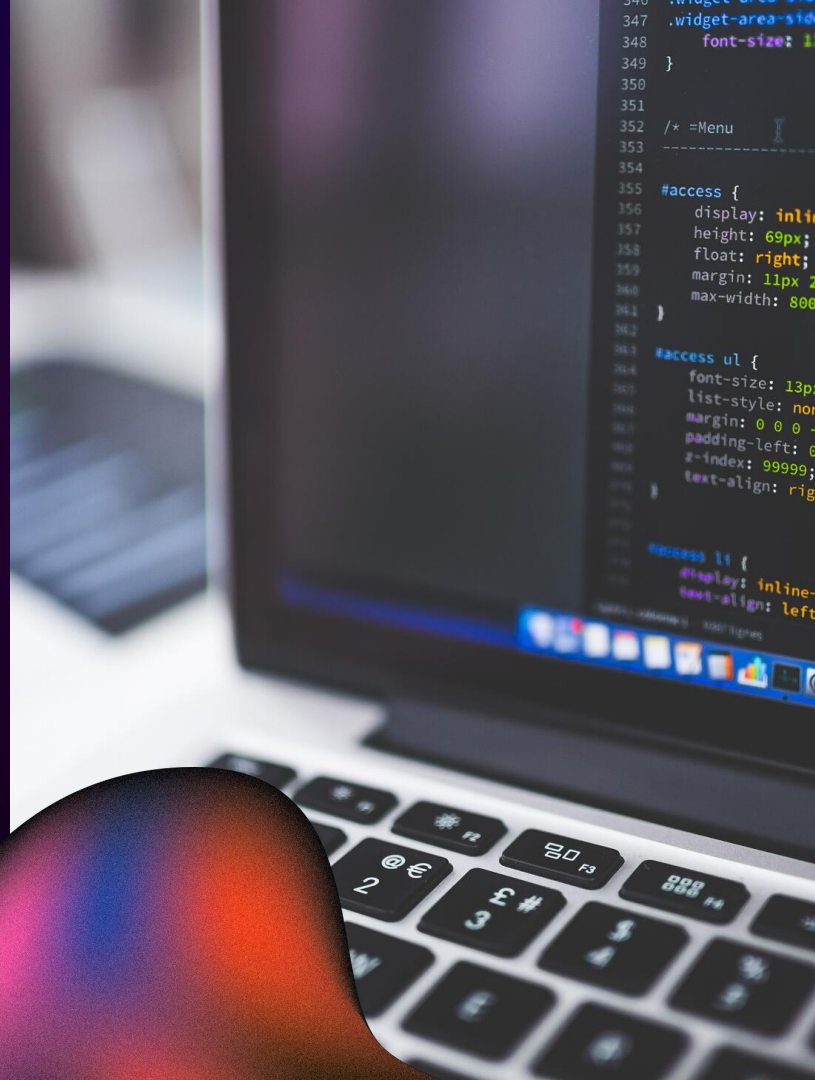
- The more questions, the better
  - Getting clarification from the customer
- Constant Communication
  - Getting input on what the team has completed and going to do

Teamwork:

- Clear roles/Responsibilities
  - Doing their part without messing with someone else's work
- Handling Different Working Styles
  - Some work at different rates/times than others

Non-Technical:

- Time management/prioritization
  - Making sure the team is on track and not wasting time on non essensial objectives
- Adaptability
  - Change is always going to happen, act accordingly and finish what is needed



Team 6

Leveraging Unity for Advanced Scientific Data Visualization and Interaction in Virtual Reality (LUADV – VR)

Ìdùnnúolùwà Adénìjí, Maria Santos Perez, Habiba Morsy, Kiara Bowen

HIDDEN FIGURES 2.0
TEAM NUMBER 6

LUADV-VR Project Overview

- **Summary:** This project develops an interactive and adaptable Unity platform for visualizing large-scale datasets in Virtual Reality (VR), aimed at enhancing data exploration.
- **Problem Domain:** 2D/3D tools lack immersive and interactive capabilities to explore complex dataset. Scientists face challenges when exploring and analyzing particle datasets in 2D/3D. (Jarrett, 2021)
- **Goal:** To allow scientists visualize, interact, and manipulate large-scale particle dataset in a VR environment.
- **Users:** Scientific Data Visualization Scientists at Argonne Leadership Computing Facility.
- **Needs being addressed:** This project integrates VR within Unity to fetch and load datasets, offering VR tools for data manipulation and interaction. These tools include an opacity slider, time-lapse functionality, and scale adjustment.

Jarrett, T., Comrie, A., Marchetti, L., Sivitilli, A., Macfarlane, S., Vitello, F., Becciani, U., Taylor, A., Van Der Hulst, J., Serra, P., Katz, N., & Cluver, M. (2021). Exploring and interrogating astrophysical data in virtual reality. Astronomy and Computing, 37, 100502. https://doi.org/10.1016/j.ascom.2021.100502

Project requirements – Functional & Non-functional

- **Software used:** Unity Gaming Engine (v. 2023.2.2) – Versatile, adaptable and commonly used software in the visualization community.
- **Hardware used:** Meta Quest 2 headset & Alienware Aurora R15 with GeForce 3090 PC

FUNCTIONAL REQUIREMENTS	NON-FUNCTIONAL REQUIREMENTS
• Upload and Download Dataset: Files in .csv format. • Parse and visualize Dataset • Adjust visualization parameters: Users can adjust particle opacity, color, and size for enhanced data visualization • Select Data Attributes: Files need to have at least two attributes. e.g., Velocity and Density • Navigation • Manipulate Dataset using VR tools • Time-lapse Functionality: More than one time-step should be loaded	• Usability: interface is intuitive and easy to use • Performance: less than 10 seconds upload per timestep • Compatibility: Windows, macOS • Scalability: handle increasingly large datasets without significant performance loss • Reliability: system should function consistently and recover from any errors without data loss

PRODUCT BACKLOG

Sprint 1

- Delivers core functionality for data upload, parsing, and basic visualization.

Sprint 2

- Expands visualization features, including adjustable parameters and data attributes.

Sprint 3

- Focuses on user interaction with the dataset in VR, including navigation, manipulation tools, and time-lapse functionality.

Lessons learned - so far

Project

1. Adding debug statements within methods to easily identify which part of the code fails during execution.
2. Reviewing Unity's documentations to identify deprecated methods and their alternatives.

Client's feedback

1. Prioritize the development of download and run functionalities rather than focusing on upload functionality.
2. Ensure files are sorted during upload, as the order is crucial for a smooth transition in time-series.
3. Utilize arrays to store multiple files in the backend development.

Teamwork

1. Setting clear roles and expectations among team members helped streamline development process.
2. Importance of turning in task on time by setting internal deadline as a team.
3. Reviewing progress regularly with the client helped align the project with the user need.
4. Regular communication is crucial for effective collaboration and overcoming challenges.

Team 7

ByteSized Solutions

Team Number: 7

Team Name: ByteSized Solutions

Team Members: Essence Toone, Janai Campbell, Kimberly Opara, Shavone Taylor, Ishan Patel

Project: ScheduleMate

Everyday Problems

Inefficient Scheduling: Managers struggle with manually trying to assign tours without having to individually check each colleague's availability, leading to time-consuming manual processes.

Scheduling Conflicts: Overlaps in tours can occur, especially when multiple guides or speakers are involved, causing confusion and logistical challenges.

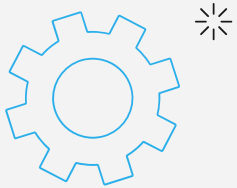
Inconsistent Data Tracking: There is a lack of systematic tracking for tour frequency.

Communication Gaps: There are challenges in keeping all parties informed about tour schedules, reminders, and changes, leading to missed assignments and disorganization

Manual Assignments: The reliance on manual input for assigning tours creates inconsistencies and may lead to unfair distribution of workload among ambassadors.

Project Users and Their Needs: Managers, student ambassadors, and office staff who require an intuitive system to schedule, track, and analyze tours effectively.

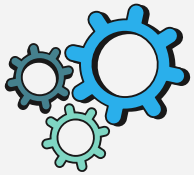
Problem Domain: Managing user roles, preventing scheduling conflicts, ensuring real-time updates, and integrating with external systems(i.e Sling).

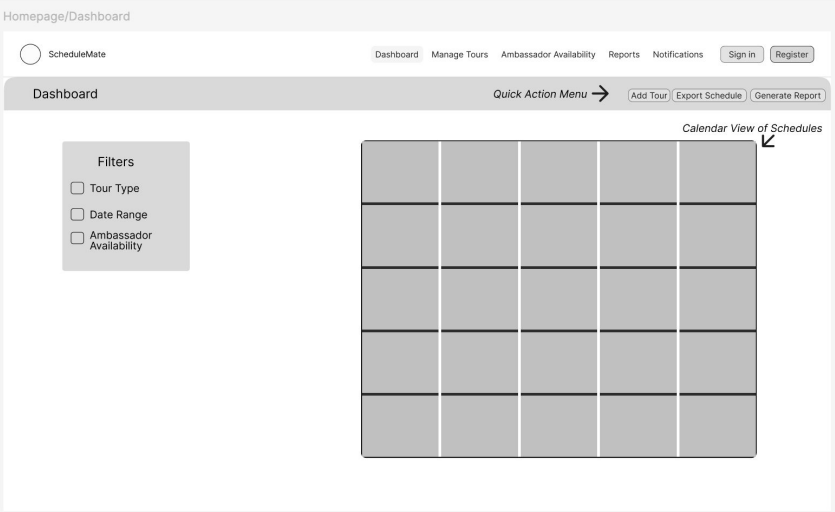


Our Solution

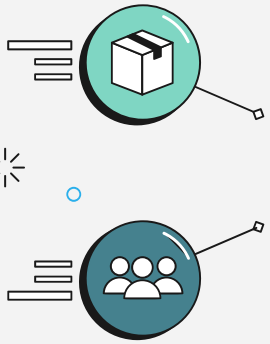
- A software that automatically schedules students ambassadors to fixed tour times based on their set availability
- Send alerts to students when assigned to roles
- Data visualization features analyzing the student ambassadors' progress and tour count
- Ambassadors can see their scheduled tours, and request to swap tours
- **Project Goal:** Make student ambassador scheduling more efficient by reducing human error with an automated system.

ScheduleMate





Tools Used and How they were Selected



Frontend Tools

HTML, CSS, JavaScript

These tools were chosen to build a responsive, user-friendly interface. **HTML** provides structure, **CSS** handles styling, and **JavaScript** enables dynamic interactivity, ensuring compatibility across both desktop and mobile platforms.

Backend: Node.js

The backend will use **Node.js** for its efficiency in handling weblogic, APIs, and asynchronous requests. It integrates smoothly with JavaScript frontends, providing high performance and scalability for the scheduling app.

Database: MySQL

MySQL, a relational database, was chosen for its ability to manage structured data, such as users, events, and schedules. It offers robust querying, data persistence, and scalability, making it well-suited for handling complex scheduling data and relationships within the system.

Version Control & Project Management

Git/GitHub is used for source control and team collaboration, allowing easy tracking of progress and code reviews.

Agile methodology ensures iterative development and continuous feedback, with **Jira** used for managing tasks and milestones efficiently.

Functional Requirements & Client Feedback

Functional Requirements: Schedule creation, role-based access, and data analysis/reporting.

Non-Functional Requirements: Performance, scalability, user-friendliness, and security.
System Delivery: The fully functional system will be delivered to the **Office of Admissions**, meeting both functional and non-functional requirements.
Concluding our most recent meeting with our client, we were able to gather the minor details of the user stories that she wanted to add and/or change.

- **The client doesn't want the software to prevent the system from scheduling ambassadors if they exceeded 3 tours, rather, send her an alert**
- **The client mention that it's not a priority to have data visualizations of the recurring schools that visit**
- **The client expressed the importance of tracking tours by semester to allocate the resources efficiently during peak tour periods**
- **The client confirmed that all tours are set times and dates.**
- **The client wants the software to allow an option for a new ambassador to shadow an experienced one, expressing the importance of the software being able to differentiate between experienced and new ambassadors**
- **The client wants the software to KU colors and themed**

This feedback allows us to design the software specific to her details so we adjusted the user stories based on her comments.

Sprint Planning and Product Backlog

Project Progress: ByteSized Solutions is on track to complete the project by December. We've followed a structured timeline, with key milestones being met, and have 7-8 weeks left until final delivery in the first week of December. Each sprint is scheduled to last 2 weeks, with the final week reserved for deployment and testing.

- **Sprint Highlights**:
  - **Sprint 1** focuses on building the core scheduling engine.
  - **Sprint 2** enhances the system with specific client requirements, such as notifications.
  - **Sprint 3** adds user access controls and data analysis features.
- **Backlog Management**: User stories were prioritized based on client needs, ensuring that the most critical functionalities were addressed first. Each sprint targets key system features like the scheduling engine, notifications, and reporting tools.
- **Final Delivery**: The final system will be fully operational, thoroughly tested, and delivered at the end of the sprints, ready for client use in December.

Client Turnover Process & Lessons Learned

Client Turnover Process

- Final System Delivery - The fully operational system will be delivered to the Office of Admissions, ensuring all features are tested, functioning as expected, and ready for use. This includes a detailed walkthrough of the system to confirm that all client requirements have been met.
- Training - Hands-on training for managers and student ambassadors to learn the system functionalities.

Lessons Learned (interacting with the customer, teamwork, and other non-technical aspects of the project)

1. **Reflection**: Analyzing what went well and what didn't during our zoom meetings on Sunday for the previous week
2. **Documentation**: Insight to create and develop software under someone else's liking.
3. **Continuous Improvement**: Using client feedback to refine practices, enhance team performance, and improve outcomes and expectations.
4. **Adaptability**: Being willing to adjust strategies and take constructive criticism.