CS 361 Course Project
CP10 Deliverable
ProgressAD
Requirements Specification Document
December 6th, 2022

Alyssa Kittle, kittlea@oregonstate.edu
Tom Nyuma, nyumat@oregonstate.edu
Daniel Zahariev, zaharied@oregonstate.edu
Logan Helstad, helstadl@oregonstate.edu
Austin Cash, cashau@oregonstate.edu
Jonathan Rockett, rockettj@oregonstate.edu
John Rundle, rundlej@oregonstate.edu

Table of Contents

1 Introduction

1.1 Purpose of Document

The purpose of this document is to form the basis of the organization of our course project, ProgressAD. The requirement specifications within this document describe and expand on all technicalities of what's needed from the product such as functional and non-functional objectives, context model, static model, and class model. However, for the scope of this document, we will not define how to deliver this application, or how to build what is needed. Although, we will define and lay the groundwork for what the software needs to do and how it must perform.

1.2 Project Summary

Project Name*:*	*ProgressAD*
Project Page:	***https://trackmedixon.onrender.com/***
Github Repository:	***https://github.com/TrackMeAtDixon/Progress***
Demo Video:	**https://youtu.be/B000yfQR0xM**
Teaching Assistant*:*	*Kunal Rathore*
Team Leader*:*	*Tom Nyuma*
…….	
Responsible Analyst*:*	*Kunal Rathore*
Responsible Users*:*	*Dixon Recreation Center Guests*
Responsible Tester*:*	*Kunal Rathore*
…….	
Team Member	*Daniel Zahariev*
Team Member	*Logan H Helstad*
Team Member	*Austin J Cash*
Team Member	*Jonathan S Rockett*
Team Member	*John Walker Rundle*
Team Member	*Alyssa Ann Kittle*

1.3 Project Scope

ProgressAD will be self-contained within the Dixon Recreation Center, without any influence from outside of or even other systems within this location. The only influence that will be allowed is the users of this application, people who are attending the Dixon Recreation Center, who are choosing to interact with ProgressAD.

1.4 Background

Currently, there is no system in place at the Dixon Recreation Center that functions as or resembles a workout journal. With the Dixon Center containing all of Oregon State University's recreational equipment (sport equipment, workout equipment, courts, pools, etc.), a lot of the university's students will come to this location to exercise in some way. These students also tend to have busy schedules and hold a lot of pressure from their academics, which makes it difficult to maintain or plan a rigorous and consistent workout schedule. When attempting to improve one's own body with a gym, it is paramount that workouts are consistent, otherwise progress can be very slow.

Therefore this new system, ProgressAD, will provide the missing piece to allow all Dixon Recreation

Center users to make steady progress while achieving their workout goals that will do all of the hard work in setting up workout routines and keeping track of all of their workout data that they are able to look back on. Since this application will fill-in as a workout routine planner, workout journal, and workout calculator without needing much user-interaction will allow more users to come to the Dixon Recreation Center on a consistent basis. Providing much more traffic to the Dixon Recreation Center to take advantage of all the equipment that is offered.

There are similar systems on the market that function as a workout journal and workout companion applications that contain everything the basic features ProgressAD has. However, these applications aren't focused around the Dixon Recreation Center and are more of general journaling, which makes it need more user interaction and knowledge to be able to document their workout data in an efficient manner. But with ProgressAD, since the application is developed solely around the Dixon Recreation Center it needs less user interaction to determine what machine a user will use and the data they would like to see.

1.5 System Purpose

The purpose of our system is for students, university faculty, and anyone else who attends Dixon Recreation Center to have a safe online workout journal where they can easily track and view their workout progress. Since the system will be based and used in the Dixon Recreation Center within Oregon State University.

The system will be used to log in a user, create a workout (cardio or strength), and mark machines as in use or available. Once the user completes their workout they can log their data into certain inputs and then see their progress on different types of graphs.

It can be difficult to motivate yourself to stick to a workout regiment, especially when you aren't seeing any visual progress being made on your body. Being able to visualize on a graph what progress you've made for your health is a great way to stay motivated to keep working out and going to the gym. The system will not be responsible for recommending new workouts.

1.6 Document Overview

The document is organized in a series of headers, with the functional and non-functional objectives being first. Later on in the series of headers/end of the document, the document will introduce a system design which visualizes our many diagram structures such as our class diagram, context diagram, and sequence diagrams.

2 Section: Functional Objectives

2.1 User Login with Correct Values

A correct user login will redirect the user to ProgressAD homepage.

Name	User logins with correct username/PIN
Requirement	When the user logins with their valid username and PIN, the system shall process their information and allow the user to get access to ProgressAD.
Rationale	Users need to be able to have a unique account that is linked to themselves that they are able to access as all features in ProgressAD are only concerned with the user's information, no other user's data will be used when a specific user is logged in.
Priority	High
Priority Reason	The user being able to login into ProgressAD allows each user to be uniquely identified to be able to save their workout data into the database and get their historical data when needed by the system (show data into graphs and show past workouts).
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.2 User Login with Incorrect Values

An incorrect user login will continue showing the login page but with errors on the specific field.

Name	User logins with incorrect username/PIN
Requirement	If the user tries to log in with an incorrect username and/or PIN, then the system shall reject the login information and ask the user to retry login.
Rationale	Trying to login using the incorrect information should not let the person log in. (ie, if someone tried to login under the username Rob34 when their actual username is Rob43, they shouldn't be able to get in.
Priority	High
Priority Reason	Being able to login using incorrect credentials opens up a very serious security breach. People could log into other peoples accounts easily.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.3 User Registration

A user will have the capability to register for an account for ProgressAD.

Name	User registers with their information
Requirement	When the user begins their registration, the system shall redirect them to the registration form to allow the user to create their unique username and PIN and to also input their body's information.
Rationale	For a user to begin using ProgressAD, they will need an account and they should have the ability to create their own account without an admin or an outside interaction.
Priority	High
Priority Reason	All users that use ProgressAD will need to be able to be identified with their account, so that ProgressAD will be able to provide historial workout data based on their account information.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.4 User Registration With Existing Information

A user that tries to register for an account which another account has the same username and/or PIN, it will redirect the user to register with a different username/PIN.

Name	User registers with existing username/PIN
Requirement	If the user tries to register with an username and/or PIN that is already associated with another account, then the system shall reject the registration information and ask the user to retry registration.
Rationale	To keep users from being able to mix-up their account information, verifying that a new account that will be registered will contain the same username/PIN will make sure that all accounts are uniquely identified and that ProgressAD won't be able to mix-up different account data.
Priority	High
Priority Reason	Accounts shouldn't be able to contain the same username and/or PIN so that all accounts are separate and that ProgressAD is able to pull information from one account without accidentally pulling information from another account based on their username.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.5 User Starts a Workout

Once a user wants to start a workout, they will be able to click on the start workout button that initiates a new state for the account that allows them to add exercises/machines to their workout.

Name	User starts working out
Requirement	When the user starts their workout by clicking on the "Start Workout" button, the system shall redirect the user to the workout page and change state that allows the user to add exercises/machines to their workout.
Rationale	ProgressAD needs a way to determine when a user is working out and when to allow them to begin adding exercise and machines to their workout. Additionally, ProgressAD needs to keep track of when a user begins their workout therefore, a button that clearly states when a user starts to workout gives ProgressAD this information.
Priority	High
Priority Reason	To be able to allow a user to add exercises/machines to their workout needs to be controlled and ProgressAD needs to keep track of when a user begins their workout, without this feature ProgressAD won't be able to keep track of how long a user is working out nor be able to determine that they are adding exercises/machines while they are working out.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.6 User Adds Machines

A user will be able to add machines to their workout that they are using (treadmill, strength machines, etc.).

Name	User adds machines to their workout
Requirement	While a user has started their workout, the system shall allow the user access to the workout page and allow the user to add machines to their workout.
Rationale	While a user will want to be able to view the machines they used in the past, ProgressAD needs to be able to determine if a machine at the Dixon Recreation Center is being used to let other users know if a machine they are wanting to use is unavailable.
Priority	High
Priority Reason	A user adding the machines they are using is related to another feature of ProgressAD that displays what machines are being used or not, by allowing a user to add the machines they are using in their workout will provide that input for that other feature.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.7 User Adds Exercises

A user will be able to add exercises to their workout to allow them to keep track of all the exercises they have done.

Name	User adds exercises to their workout
Requirement	While a user has started their workout, the system shall allow the user access to the workout page and allow the user to add exercise to their workout.
Rationale	For ProgressAD to function as a workout journal, it will need a way to keep track of the exercises a user has done. Therefore, allowing a user to add each exercise they have done will provide a list to save for the user to look back on.
Priority	High
Priority Reason	A core part of ProgressAD is for a user to be able to journal their workouts so they are able to look back at these past workouts and visualize what their workouts looked like before.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.8 User Connects Exercise to Machine

If a user does an exercise that involves a machine, they will be able to connect that exercise with that machine (sets, reps, distance, length, etc.).

Name	User adds an exercise and connects it to a machine in their workout
Requirement	When a user has added a machine and when a user then adds an exercise to their workout, the system system shall allow the user to be able to connect that exercise to a machine.
Rationale	Exercises are independent from machines and some exercises are able to be completed without a machine, however, machines aren't independent and the user must provide their exercise information for a machine they have used.
Priority	High
Priority Reason	To be able to add information that was done on a machine is crucial to be able to save that data to compare different workout information.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.9 User Chooses State of Machine Availability

Once a user adds a machine to their workout, it is assumed that they will be using that machine right away and the user then will be able to end use of that machine. They will also be able to reuse that same machine the same way.

Name	User can end use of machine and reuse machine
Requirement	When a user has added a machine and when the user is done with that machine, the system system shall allow the user to change state of use on that machine by clicking the slider within that specific machine's card.
Rationale	Users will need to be able to let ProgressAD know that they are done working out on a machine so that ProgressAD can change the availability for other users to see.
Priority	High
Priority Reason	ProgressAD needs to be able to keep an up-to-date status of all machines that are used in the Dixon Recreation Center so that all users are able to see which machines they want to use are available or not.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.10 User Ends a Workout

Once a user is done with their workout, they will be able to end their workout by clicking on the "End Workout" button and rate their workout (tiredness and effort levels).

Name	User ends their workout
Requirement	When the user ends their workout by clicking the "End Workout" button, the system shall ask the user to rate their workout given their tiredness and effort levels throughout the workout and redirect the user back to the home page.
Rationale	ProgressAD will need to keep track when a user ends their workout to save the length of their workout in the database and also allow the user to rate their workout so that it can be saved to be viewed at a later time.
Priority	High
Priority Reason	ProgressAD needs to know when the workout ended so that the system is able to calculate how long the user worked out for and also receive the workout ratings from the user that can be later compared to other workouts the user does.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.11 User Views Reports

A user will be able to view their historical workout data in a variety of graphs based on the information that needs to be displayed (types of machines used, days they have worked out, workout duration, etc.).

Name	User views their workout reports
Requirement	When a user clicks on the “REPORTS” button, the system shall redirect the user to the reports page that will display all past workout data to the user in a variety of graph formats.
Rationale	Users of ProgressAD will need to be able to look at past workout information they have been storing within the system.
Priority	High
Priority Reason	A core feature of ProgressAD is being able to visualize past workout data for the user so that the user is able to compare their workouts and also see their progress throughout their workout journey.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.12 User Views Machines

A user will be able to see which machines are available or not in the Dixon Recreation Center when they head to the machines page.

Name	User views machines
Requirement	When a user clicks on the “MACHINES” button, the system shall redirect the user to the machines page that will display machines that are at the Dixon Recreation Center and also the availability of this machine (if another is using this machine or not).
Rationale	For this application to be built around the Dixon Recreation Center, it will need to be able to keep track of which machines are in use or not for other users to see. This will allow them to determine the type of workouts they are wanting to complete at that time.
Priority	High
Priority Reason	Being able to view which machines in use will allow multiple users at the same time to be working out and not rely on the same machines to workout. Additionally it will provide users the knowledge of what machines they will be able to use if multiple users are working out at the same time.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.13 User Views Past Workouts

When a user has done workouts in the past, they will be able to see this information (machines used, exercises done, etc.) on their homepage.

Name	User views saved past workouts
Requirement	When a user clicks on the “Home” button or when the user is redirected to the homepage in other ways, the system will show the user their workout history and the machines they used, exercises done, and ratings for each workout.
Rationale	For a user to see a whole workout rather than individual values from those workouts will help them compare all their workouts.
Priority	Medium
Priority Reason	While those individual values (ratings, machine types, workout duration, etc.) are more important, it will be useful for the users to be able to see what their workouts looked like as a whole.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.14 User Views Profile

After a user registers their account, they will be able to access their account information to view and be able to change their information (weight, height, name, etc.).

Name	User views profile
Requirement	When a user clicks on the “Profile” button, the system shall redirect the user to the profile page that will display their account information and allow the user to be able to change any of this information.
Rationale	After a period of time, a user might want to change their account information to better reflect what they look like now. For example, changing of weight and age can be changed to be up-to-date with the user.
Priority	Low
Priority Reason	A user being able to change their account information isn’t necessary as that information won’t be shown in the reports and that data is mainly used to calculate the user’s BMI.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

2.15 User Logout

Once a user is finished using ProgressAD, they will need to be able to logout so that other users will be able to have access to that public machine.

Name	User logs out of ProgressAD
Requirement	When a user clicks on the "Logout" button, the system shall logout the user out of ProgressAD and redirect the user to the login page.
Rationale	If other users are wanting access to ProgressAD that is on a public system, users will need to be able to log out of their account so that other people will have a chance to log in to theirs.
Priority	Low
Priority Reason	If the ProgressAD was strictly on a public kiosk, this would have a higher priority. But, since ProgressAD is system-independent, logging out isn't the most important feature to have in ProgressAD. If implemented on a machine that was used publicly, logging out would be a high priority.
Status	Approved and validated
Contact	Tom Nyuma
Source	Kunal Rathore

3 Section: Non-Functional Objectives

3.1 Reliability

- *Maximum bugs or defect rate – usually expressed in terms of bugs/KLOC (thousands of lines of code), or bugs per function-point.*

3.1.1 Availability

100% Database Time through MongoDB's Cloud Provider, 99%+ Express Web Server Uptime/Availability dependent on Dixon's deployment hosting provider. Hours of use for public kiosks are Dixon Recreation Center's business hours. With no deployment on public kiosks, uptime is 99%+ on non-public systems.

3.1.2 Accuracy

ProgressAD's outputs will need to be 100% accurate on calculations that are made. Specifically, values that are calculated in the reports page will need to be 100% correct based on the mathematical equations that were used and BMI will need to be 100% accurate as well determined on the BMI equation.

3.1.3 Mean Time Between Failures

After a failure, there will be a mean time of 3 days until it is back in operation again.

3.1.4 Mean Time To Repair

Once the application has met a failure, the mean time to repair will be 3 days. In general, the first day will be used to determine the reason for the failure and plan the solution. The second day will be the implementation of that solution and the third day will be the testing of that solution.

3.1.5 Bugs or Defect Rate

Minor Bugs - Such as styling errors (no impact on functionality or data), will be ignored. If there are too many styling errors, that could provide the user with an inefficient application. That being said, just a few styling errors won't impact the user dramatically when using the application.

Significant Bugs - Such as routing errors or functions breaking (impact on functionality), will need to be addressed as soon as they occur as this bug should not be allowed to continue based on the number of

people who use the Dixon Recreation Center.

Critical Bugs - Such as complete loss of data or total inability to use the application, will need to be addressed ASAP and all systems that have access to ProgressAD will need to be turned off.

3.2 Usability

3.2.1 Training Time for base and power users

- Users shall need no outside training time and shall either learn to operate the application from contextual clues in the UI, or via signage posted within Dixon.
- In order for power users to become productive at tracking their workouts, they shall require no more than 6 sessions of use for separate workouts — in other words, two weeks assuming three workouts a week.

3.2.2 Base Usability

- The UI shall have a minimum base usability so that an average user who is familiar with typical U.I. conventions can access at least half of the functionality within the first user session.

3.2.3 Training Time for Admins

- Admins shall only require two weeks, (assuming 5-day work weeks) to become fully productive with the administrative functionality of the application.

3.3 Performance

3.3.1 *Response Time (Avg, Max)*

The average response time for ProgressAD will be between 100ms to 200ms.

The maximum response time for ProgressAD will be 1 second or 1000 ms.

3.3.2 *Capacity*

The total number of users that are able to use ProgressAD will be 100 users.

3.3.3 *Resource Utilization*

Database Network Connection - 1,000 B/s In and 15 kB/s Out.

3.4 Security

3.4.1 Username & PIN Security

- If and only if a user sends a correct username and password, the server shall provide access to their user-specific pages.
- The pin shall be encrypted so that no one with access to the database shall be able to interpret the password

3.4.2 Account Data Privacy

- No user data shall be sold to third-party companies.
- User data will be encrypted; the data encoder will be abstracted away from both the developers of ProgressAD, and users.

3.4.3 System Functions

- Only users with valid environment variables will be able to interact with the system internals.

3.5 Supportability

3.5.1 Code Formatting

- The code-base for ProgressAD shall follow a strict code formatting guideline ensuring that

consistency and standardization is maintained throughout the code.

3.5.2 Maintainability

- The ProgressAD official GitHub promotes the constant integration and maintenance of the system through pull-requests, allowing developers to obtain the repository, make changes, and get those changes approved by members of the application development team.

3.5.3 Scalability

- The ProgressAD web-service runs a load balancer in the API Gateway entry-point instance to evenly distribute network request traffic and support the following scaling methods:
    - Manual (Developer manually sets a number of instances)
    - Auto-scaling (Render.com automatically scales the web-service based on target CPU and memory utilization)

3.6 Design Constraints

3.6.1 Frontend Framework and languages

- The frontend application component shall be built using JavaScript and React.
- The frontend application component shall use the material design UI component library.
- The frontend component shall be a static site; bundled to be served in the browser using webpack.

3.6.2 Backend Framework and languages

- The backend application component shall encapsulate the frontend components according to Express's API building framework.
- The backend application component shall use MongoDB's unstructured data (NoSQL) to store the user, workout, and dixon machine data.
- The backend application component shall use Cloudinary media hosting services to store Dixon Recreation Center's machine pictures.

3.6.3 Devices and Running

- The ProgressAD system shall be system independent, enabling base compatibility with all modern web browsers.

3.7 Online User Documentation and Help System Requirements

3.7.1 User Documentation

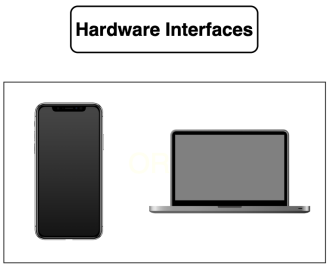
- The documentation for ProgressAD is hosted on GitHub. The documentation contains an API Reference, installation guide, instructions for getting started, MIT License, and high level technical specs.

3.7.2 Help System

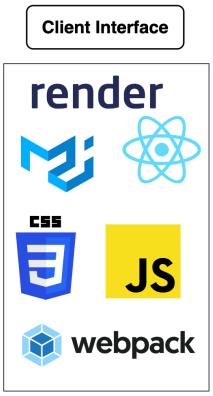
- The help system is encapsulated within the GitHub Issues system. Clients needing help or facing issues with the software can head to ProgressAD's Issue System to resolve and discuss their issues to obtain help.

3.8 Interfaces

3.8.1 Hardware Interface

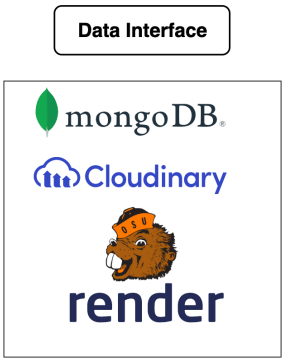


3.8.2 Client Interface



3.8.3 API Interface

3.8.4 Data Interface



3.9 Licensing Requirements

ProgressAD is using the *MIT License*. The MIT License a short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

3.10 Legal, Copyright and Other Notices

The software component contains a copyright notice to the ProgressAD system. There hasn't been any legal justification for actually obtaining the notice, but for the purpose of this document, our application does contain a non-applicable copyright.

3.11 Applicable Standards

- The ProgressAD application is system independent.
- Any device with a browser can access the ProgressAD service.
- ProgressAD is free to use and open-sourced.
- The application requires minimal-user interaction.
- The ProgressAD system follows a strict style guideline.
- ProgressAD uses material design UI components.

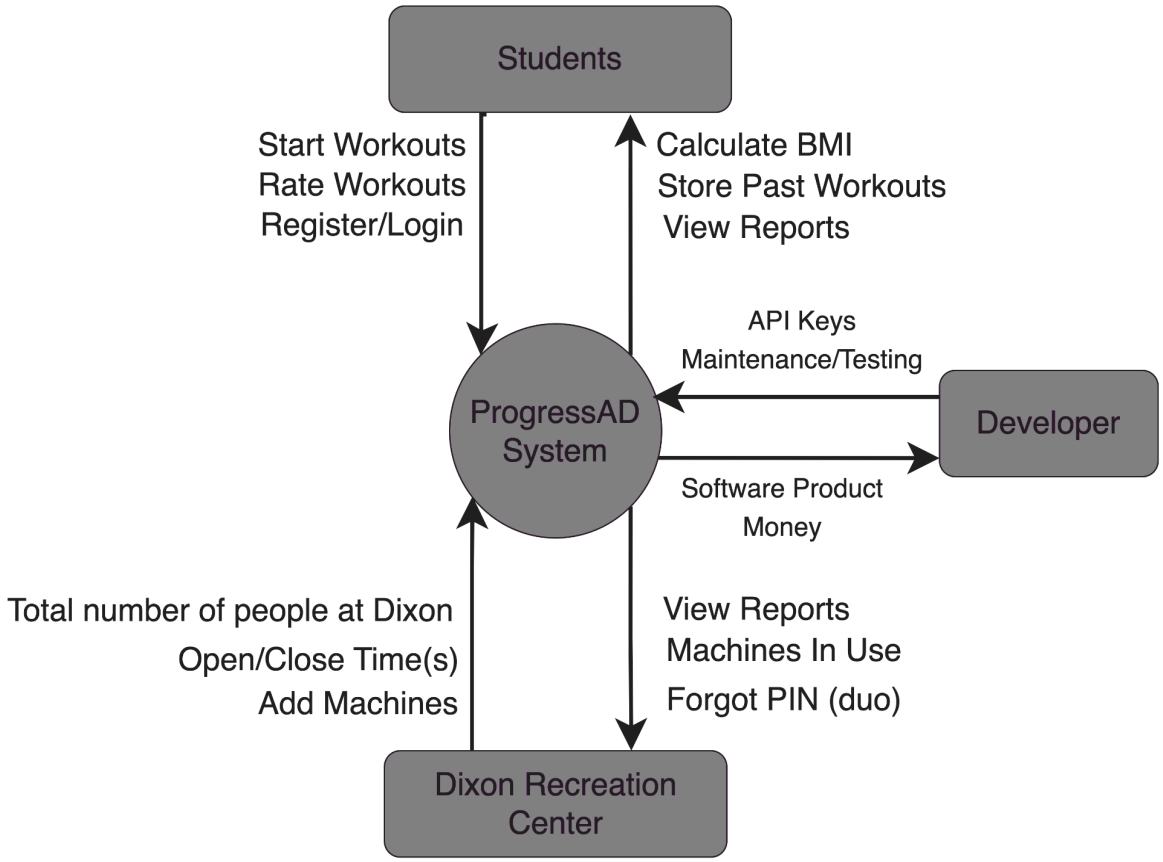
4 The Context Model

4.1 Goal Statement

Our goal is to provide a system which provides value by allowing users to keep track of workout progress. The system's high level goal is to require as little user interaction as possible.

The ProgressAD system achieves its business purpose by providing a service to Oregon State University students with an alternative to the traditional fitness tracker solutions for little to no cost, provided kiosks are not purchased.

4.2 Context Diagram



4.3 System Externals

Students - Students are the primary user of the application. Students see the front facing application, enabling them to get the main value out of the application through interactions with the software system.

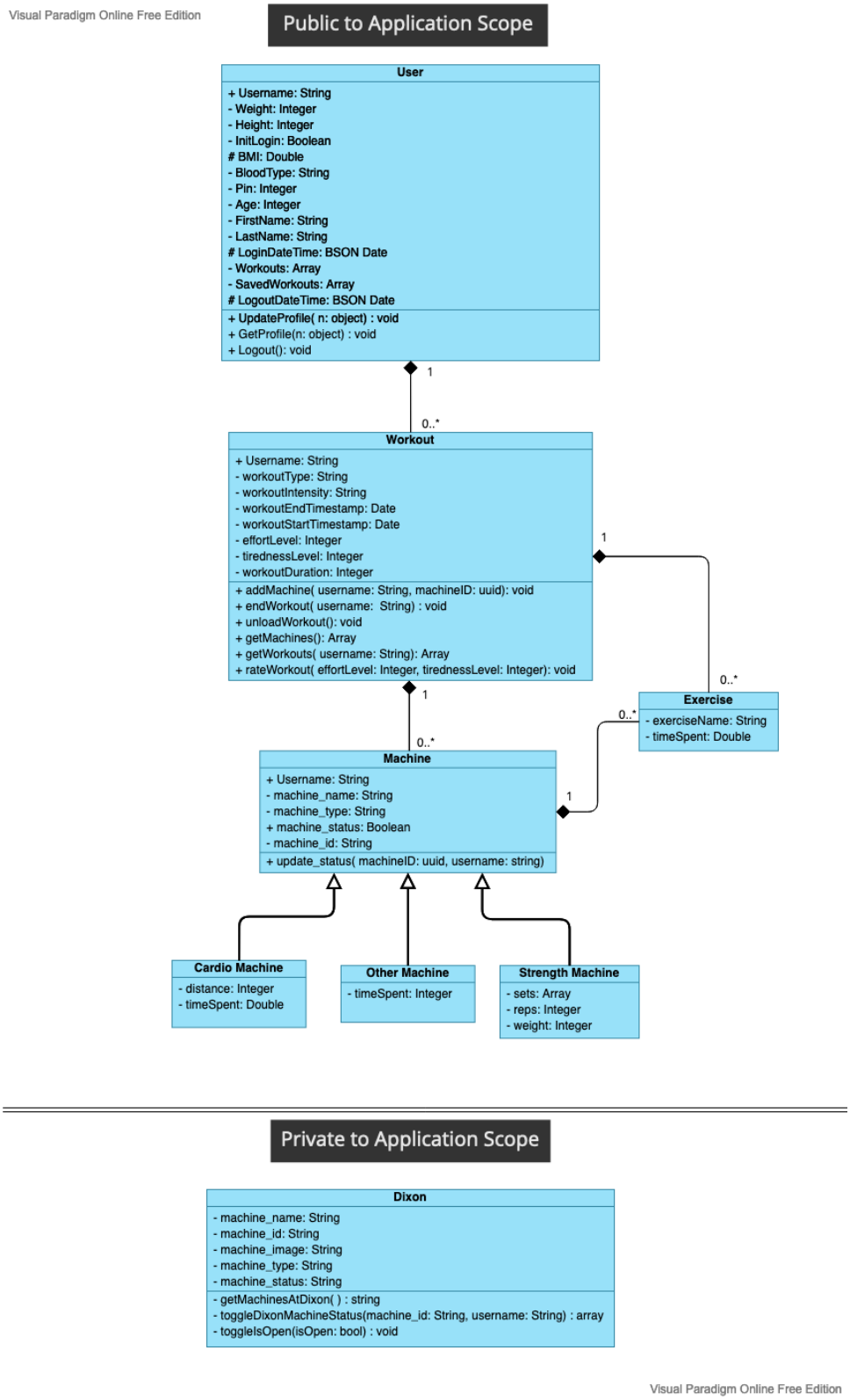
Developer - Developers are the maintainers and testers of the application. Developers make the updates to the application and test these updates before releasing the application to the public.

Dixon Recreation Center - Dixon Recreation Center provides the non-functional components of the application and Dixon-Specific data such as the hours of the facility, machines, and total number of people.

ProgressAD System- ProgressAD is the all encompassing system. The application follows a "micro-services" architecture by decoupling the business logic away from clients who interact with the different parts of the system.

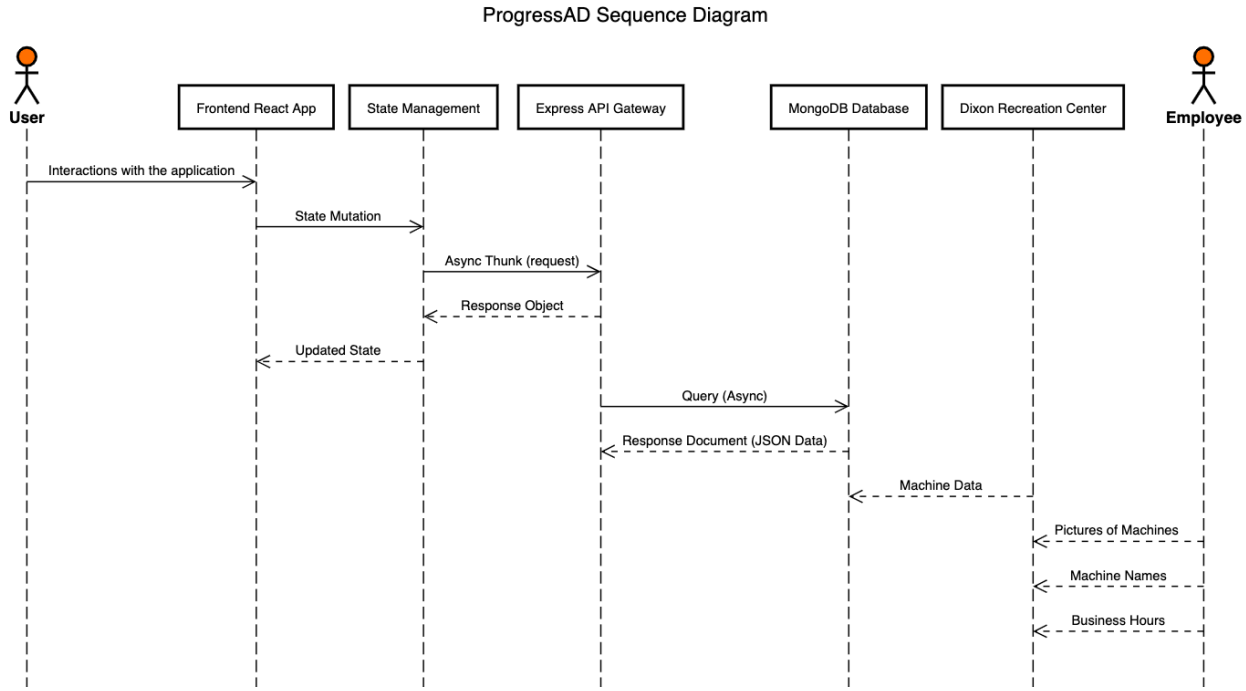
5 The Static Model

5.1 Class Diagram



6 The Dynamic Model

6.1 Sequence Diagram



7 User Acceptance Tests

User Acceptance Test 1: Workout Rating

Scenario*:* The Dixon Recreation Center user rates their workouts after being completed on a scale of 1-10, representing their current fatigue level. They log out and when they log back in they can see their past and current fatigue level ratings on a timeline.

Acceptance Criteria*:* Given that I just completed my workout, when I click the “End Workout” button I should see a survey asking me to rate my effort level and tiredness level on a scale from 0-10.

Step	Task Description	Expected Results	Status (Pass/Fail)	Comments	Tester Name
1	*While logged in from the home page, start the workout.*	A modal pops up - The first modal asks the user to select a workout type. - The second modal asks the user to select a workout intensity level.			
2	*End workout*	A modal pops up - The first modal asks the user to rate their workout’s effort level. - The second modal asks the user to rate their tiredness level. Automatically redirect to home screen			

User Acceptance Test 2: Registration

Scenario*:* When a user is creating their account his BMI is automatically calculated for them, based on the height and weight that they have input into the system.

Acceptance Criteria*:* Given that I have finished registering for ProgressAD, when I login and click the “Profile” button, then I should see my BMI.

Step	Task Description	Expected Results	Status (Pass/Fail)	Comments	Tester Name
1	*Navigate to our deployed application and begin registration*	Registration page will be shown to user			
2	*Input values for form data*	Form Data should be: - Username - Pin - Blood Type (optional) - Height - Weight			
3	*Click register*	User account is created, all data from user is saved in database. BMI is calculated and saved to the database..			
3	*Click Avatar Icon*	Menu pops up			
4	*Click Profile*	Profile form with currently logged in user’s data already appended			
5	*Adjust profile information*	BMI value is shown to user Can be adjusted by moving the height and weight sliders			

User Acceptance Test 3: Adding a Machine

Scenario*:* When a user is looking for a machine to workout on, they look at the machine page to check which machines are available and which machines are being used by other people.

Acceptance Criteria*:* Given that I am logged into ProgressAD, when I click on the “MACHINES” button to go to the machine page, then I should see which machine is available or not available.

Step	Task Description	Expected Results	Status (Pass/Fail)	Comments	Tester Name
1	*Log in, if not already logged in*				
2	*Click on machines*	Machine page is shown to user			
3	*Look at all machine statuses*	All machines’ statuses are visible to the user			
4	*Start workout*				
5	*Add machine to workout*	Machine status will be unavailable to other users			
6	*Click on machines*	Machine page is shown to user			
7	*Check machine that is in use for workout*	Machine status for the toggled machine will be unavailable			

User Acceptance Test 4: Historical Workout Data

Scenario*:* When a user has historical workout data (multiple machine type use, multiple days they have worked out, multiple effort and tiredness values, etc.) they are able to check this data in a graphical format when they click into their reports page.

Acceptance Criteria*:* Given that I have past workout data, when I click the “REPORTS” button, then I should be able to see this data in a variety of graph formats.

Step	Task Description	Expected Results	Status (Pass/Fail)	Comments	Tester Name
1	*Log in if not already*	Navigate to the home page			
2	*Click on “Reports” in the nav bar*	Reports page will be shown to user			
3	*Look at the 6 graphs*	Historical workout data is represented on 6 unique graphs			

User Acceptance Test 5: Adding Exercises

Scenario*:* When a user begins their workout, they are adding exercises that are either independent from a machine and/or is able to be connected to a machine they are wanting to use.

Acceptance Criteria*:* Given that I have started a workout, when I add an exercise, then I should be able to connect that exercise to a machine, input the corresponding values that are needed for that machine type, and input the exercise name.

Step	Task Description	Expected Results	Status (Pass/Fail)	Comments	Tester Name
1	*Sign in to a valid account and navigate to homepage*				
2	*From the home page, click the "Start Workout" CTA. Progress through the "begin Workout" flow*	User is taken to the workout page			
3	*Click "Add Machine"*	"Please Select a Machine." modal pops up			
4	*Select a machine and click "Add machine to workout"*	The machine is added to the workout list. The machine is now "in use"			
5	*Click "Add Exercise"*	The previously added machine appears as an option in the popup modal.			
6	*Click the machine and click "next"*	A context-appropriate modal associated with the machine appears and allows the user to enter appropriate data.			
7	*Enter data for machine and click "submit"*	Data is saved under the machine's card in the Workout page.			

User Acceptance Test 6: Previous Workouts

Scenario*:* After a user has finished their workouts, they are able to see their previous workouts on the home page.

Acceptance Criteria*:* Given that I have previous workouts that I have finished, when I go to the home page, then I should be able to see all these previous workouts I have done.

Step	Task Description	Expected Results	Status (Pass/Fail)	Comments	Tester Name
1	Log in or be logged in as a user with prior completed workouts			If your account does not have any prior completed workouts, you will need to complete at least one workout to perform this test.	
2	Navigate to the home page				
3	Observe prior workout card(s).	Card should contain the following: ● Date ● Type ● Intensity ● machines used ● Effort level ● Tiredness Level The above information should match the previous workouts			

User Acceptance Test 7: Weight Change

Scenario*:* A user wants to change their weight after their creation of their account.

Acceptance Criteria*:* Given that I have an account for ProgressAD, when I go to the profile page, then I should be able to see the weight that I last inputted was and be able to change that value.

Step	Task Description	Expected Results	Status (Pass/Fail)	Comments	Tester Name
1	*Click upper right corner icon.*	A drop down list will appear with Home, Profile, Workout, and Logout on it			
2	*Click on the profile option.*	This will take you to your profile page, where you will see your Username, Firstname, last name, age, height, and weight. Some of these values will be empty if this is your first time using this page (or if you haven't entered a first/last name or age).			
3	*Use the slider under the Weight (lbs) label to change your weight to your desired input.*	By moving the slider the number to the right of the slider will change, increasing in amount as the slider moves to the right and decreasing in amount as the slider moves to the left. As you move the scale, your BMI will also be changing (BMI is calculated in real time as you slide the scale).			
4	*Press the save button at the bottom of the screen.*	Pressing the save button will change your previous weight to your newly imputed weight value. This will also save your new BMI.			

8 Appendix

8.1 Glossary

ProgressAD*:* Application title of the software system. This document refers to the TrackMe@Dixon title given originally by the Client as “ProgressAD”, the altered title for our software system we developed throughout the course project.

State Management*:* The way our user-facing, frontend application component manages data needed in order to render the many visual components. For instance, a counter button will state an integer state variable called “count” and a boolean state variable “isClicked”.

Environment Variables*:* Variables embedded into our backend application, allowing it to behave differently based on the environment it is executing. Environment variables provide/store sensitive data like API keys, passwords, and port(s) which might change between different application instances.

Express API Gateway: Entry point of the backend, non-user facing application component. The gateway allows us to create and expose a REST API to communicate as a client with our server application, enabling create, read, update and delete operations on our application’s object model. The API gateway encapsulates the M(Model) and C(Controller) parts of the Model-View-Controller design pattern in our application.

8.2 References

(1) [ProgressAD GitHub][https://github.com/TrackMeAtDixon/Progress]
(2) [React.js][https://reactjs.org/]
(3) [Redux Toolkit][https://redux-toolkit.js.org/]
(4) [Render][https://render.com/]
(5) [Express][https://expressjs.com/]
(6) [MongoDB][https://www.mongodb.com/what-is-mongodb]
(7) [Material-UI][https://mui.com/]
(8) [Mongoose][https://mongoosejs.com/]
(9) [Webpack][https://webpack.js.org/]
(10) [Cloudinary SDK][https://cloudinary.com/documentation/node_integration]