

# UniBiz

Intern: Obafemi Oduola

Intern Manager: Qi Meng

Intern Director: Carl Taylor

Peer(s): Winston Grocholski , Daniel Lo Nigro

GitHub Repository Link: https://github.com/0bafemi/UniBiz.git

## Overview

My project aims to create a web app for campus communities that allows for connections between students and student businesses offering services like hairdressing, cooking etc. UniBiz aims to make life easier for students by letting them find and book essential services like haircuts and meals from student businesses right on campus. No more hunting through endless online listings or making trips off-campus. It helps student businesses grow by making them more visible and accessible to students, while also ensuring students can trust the services they find, thanks to verified reviews. The app encourages students to connect and support their local community by sharing their experiences through ratings and reviews.

* Category: Social Networking, E-commerce,
* Story: UniBiz will allow students to sign up with their campus credentials, browse listings of services offered by nearby businesses, and book appointments as well as directly message businesses through the platform. Businesses can create profiles showcasing their services and availability. Students can rate and review services they've used, fostering a community-driven approach to recommendations.
* Market: UniBiz is primarily aimed at university students and young adults within campus communities who seek convenient access to essential services without leaving their campus environment as well as student entrepreneurs
* Habit: UniBiz is designed to be used intermittently based on student needs, most likely times such as weekends or before major campus events or even daily for home-cooked meals. Students may use it whenever they require services like haircuts or meals, making it a valuable tool in their campus life.
* Scope: Initially, the app will focus on facilitating connections for services like hairdressing, cooking, and possibly other essential campus-related services. It will allow for profile creation, service listings, booking functionalities, and reviews. Out of scope initially are services not directly related to campus life, such as extensive retail offerings or non-essential luxury services.

## Product Spec

Based on the app description, this section goes into more detail about what the app should do, and what functionalities it must provide to the users.

### User Stories

User stories are actions that the user should be able to perform in your app.

First, focus and identify functionality that is required for your MVP (Minimum Viable Product) that conforms to all the project requirements and expectations. Make sure your technical challenges are part of your MVP.

You should also identify optional / nice-to-have functionalities that would be done as stretch goals during MU Week 8 and 9. Remember, *technical challenges should not be optional features*, they must be code complete before the end of Week 8!

#### Required

* User can sign up with their campus credentials.
* User can browse listings of local services offered by businesses on campus.
* User can view details of services, including pricing, availability, and reviews.
* User can book appointments for services directly through the app.
* User can rate and review services they have used.
* Business owners can create profiles for their services, including photos, descriptions, and availability.
* Business owners can manage their service listings and appointment bookings.

#### Optional

* User can receive notifications about new services or special offers from businesses.
* User can search for services based on specific criteria (e.g., location, service type).
* User can see a map view of businesses offering services on campus.
* Business owners can update their profiles and service information.
* User can share their booked appointments or recommended services with friends.
* User can earn loyalty points or rewards for frequent bookings.
* Business owners can offer promotions or discounts through the app.

### Screen Archetypes

[Describe the different screens that, together, compose the full experience of your app. You can leverage anything you want, such as diagrams and mocks.]

[Using diagrams you can also describe how navigation and presentation of these screens will work on a high-level.]

[These are just high-level representations though. Don’t spend too much time building mocks.]

### Data Model

[Describe the data you’re going to need to back your application. This can include database models (like tables), or external data you’ll require from some API.]

### Server Endpoints

[Describe the endpoints that your application is going to consume from your server. If you’re using REST, then you’ll probably want to include the method (GET/POST/etc) and the expected parameters (query parameters, body parameters, etc.)]

### Navigation

## Project Requirements

[Based on the [Project Guide](https://docs.google.com/document/d/1TvGTVGsH0b3HSVh_tRvQZDizWwBSQVCfRiS4sqMZY6Y/edit#heading=h.92pf0mt6mhlq), describe how your project is going to be fulfilling each of the base project requirements.]

## Technical Challenges

For your project, you should demonstrate that you can apply what you’ve learned so far and expand on that knowledge to write code and implement features that go beyond the scope of the projects you worked on during CodePath.

Based on the general idea and direction of your project requirements, your intern manager will create at least two (2) Technical Challenges for you. This section is all about explaining what they are and how you’re planning to tackle them - you’ll work together with your manager to fill it out.

### Technical Challenge #1 - [Name/Small Description]

#### What

What problem are you solving, and what parts go beyond what you learned in CodePath?

#### How

Explain in words how you’ll solve this problem.

You’re encouraged to expand on this section with pseudo-code, links to external frameworks, architecture / design diagrams, anything that you can use to explain this to others!

### Technical Challenge #2

#### What

#### How

### Database Integration

[Describe what you are using for database storage. For example, Parse, MongoDB, Sequelize, etc.]

### External APIs

[Describe at least one external API you’re using for your project. For example, Google Maps, Spoonacular, OpenWeather, etc.]

### Authentication

[Describe how user authentication is handled for your project, including logging in and signing up. Also describe any kind of cookie / session management you’re doing and how you’re implementing it, and how this affects navigation between different screens by the same user.]

### Visuals and Interactions

[Provide details on how your app is fulfilling the following UI craft requirements, and how these are technically accomplished.]

* Interesting Cursor Interaction
* UI Component with Custom Visual Styling
* Loading State

## Timeline

Project execution will start in Week 4 of MU. Based on the previously defined requirements, user stories and technical challenges, use the following table to scope out and plan a timeline for deliverables over Week 4 - 9. You can be as detailed as you need, ranging from simply mentioning the user stories, or dividing them into sub-tasks.

You are free to modify the table, add / remove rows or columns, whatever fits your style! The important thing here is that you focus and prioritize certain aspects of your project so you don’t get behind and are ready to deliver the MVP - remember your required features should be code complete before the end of Week 8, including both technical challenges!

We also encourage you to leverage project tracking tools such as GitHub Issues or Meta’s internal Tasks / GSD tooling to keep manage individual units of work.

| MU Week | Project Week | Focus | User Stories |
| --- | --- | --- | --- |
| 4 | 1 | Focus on the components that will serve as the skeleton of your project. You will probably be using most of what you learned in CodePath to set up things like the client and server repositories, initial routing, login / registration, creating a database with object models, etc. | Example:   * User can login * User can create an account * [Optional] User passwords are encrypted in the database for security |
| 5 | 2 | Week 5 and 6 should be where you focus on the specific requirements of your project. | Example:   * User can create / edit / delete posts * User can chat with other users in real-time (e.g. technical challenge) |
| 6 | 3 | By this point, you should be getting started with your technical challenges as well. |  |
| 7 | 4 | You should focus on finishing your MVP and core requirements. By this point, you should be done with at least one of your technical challenges. |  |
| 8 | 5 | Continue work on finishing touches and stretch goals for your MVP. By this point, your core functionality and both TAPs should all be in place. It is also a good point to start working on stretch goals that could further expand on the functionality (and technical complexity) of your project.  This week you also have to submit your self-review, make sure you allocate enough time for this alongside your final submission for your project! |  |
| 9 | 6 | It’s time to show others what you have built! Work on a presentation and demo that you will present to other interns to showcase your work. You are also free to continue polishing and expanding on your project! |  |
| 10 | 7 | For this week, we have a bunch of extra activities prepared to give you a quick dive of what it is to work at Meta. You will find activities around using internal tools and frameworks, and even committing code to our internal repositories. |  |