BRIDGE IN TECH

AnitaB.Org GSoC20 Student Application

Student Name: Maya Treacy
Zulip: Maya Treacy GSoC20 Aspirant

AnitaB.org GSoC20 Student Application (Part 2 of 2)

General Questions

- Website: https://mtreacy002.gitlab.io/myresume/
- Github: https://github.com/mtreacy002
- Gitlab repositories related to my learning on the tools related to the project (access will be given upon request only to respect copyright and the author of the course):
  => Course: TestDriven.io - Flask-TDD-Docker
  My project repository:
  https://gitlab.com/mytravelpa01/flask-tdd-docker.git
  => Course: TestDriven.io - Flask-React-Auth
  My project repository:
  https://gitlab.com/mytravelpa01/flask-react-auth.git

Project Specific Questions

Which AnitaB.org GSoC project from this Ideas List are you applying for (please submit a separate application for each project):

BridgeInTech - Original Idea

Inspired by: Industry-led Opportunities pipeline & Mentorship System AnitaB.Org

From this point onward, Mentorship System will be referred to as ***MS*** *and BridgeInTech will be referred to as* ***BIT***

Brainstorming Ideas

Background Motivations

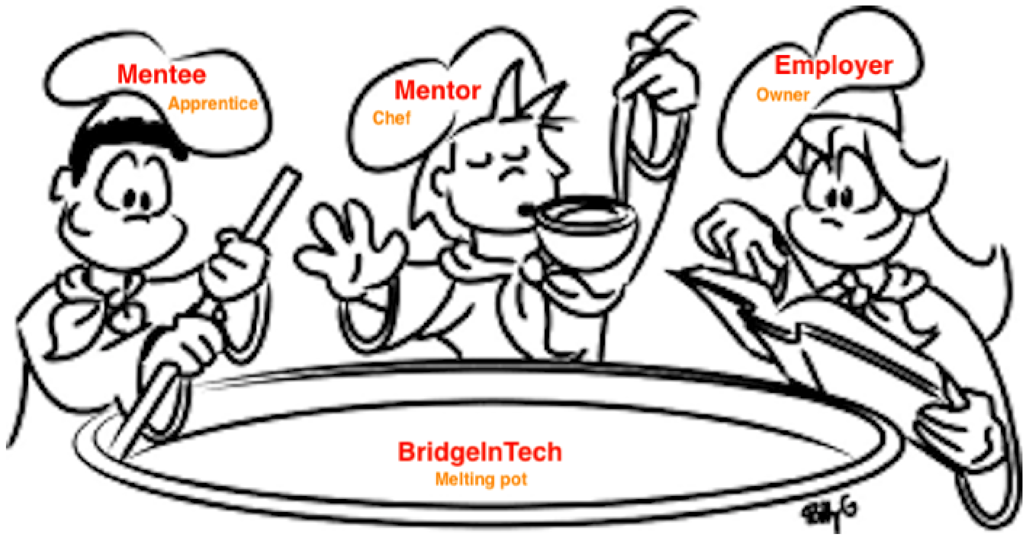


Image by: WikiArt (souce: https://images.app.goo.gl/JutinVTdeKvv4oFR9)

The idea for the app comes as the results of the following:

- the current technical skills gap in the IT industry that is felt by the businesses/employers as stated by TechRepublic, 6 June 2019 (Matteson, 2019).
- from my own personal experience, feeling like what I have learnt at university is not enough to help me be 'work-ready'. I believe the app would help newcomers like me gain further knowledge through hands-on learning from the Mentoring Program with industry affiliation.
- from self-motivation being a mature age woman who is hoping to enter the IT industry as a career change, who can be considered to be among those within the minority groups of the IT industry. This app gives a glimpse of hope for me and people who are in the similar situation of a chance to learn the necessary skills and be work-ready so we can enter the industry and grow our career.

Benefits to the Community

- It serves as a talent pool that is recognised by the industry:
    - Companies can use it as a starting point to look for a potential employee and ensure that the person will meet the skill sets they need for the similar jobs within their own company.
    - IT Professionals can use it as a chance for a career enhancement, for widening their professional network, even a possibility to gain some financial benefit (if any)

    - Graduates or other job seekers can use it as a way to gain hands-on learning to a variety of skill sets that are desirable by the industries.
- It promotes equal opportunities by encouraging companies to give priority in their mentoring program to people that fall within the minority based on the following categories: gender, race, age, sexual orientation, physical/mental ability, education, work experience, socio economic, and religion.
- Supporting AnitaB.org Mentoring System by providing industry involvement to the Mentoring Program. Company involvement can be in a form of (but not limited to):
    - offering a small project (that is too small for the company to make them hire a new person but will be too expensive for them to outsource because of the lack of talent out-there to do the job.
    - providing resources needed (such as softwares, hardwares, etc) for the Mentor/Mentee to work in the program so as to save them from having to use their own money when working on the program.
    - providing financial incentive (stipend), even though volunteering option is also available.
    - Offering support throughout the whole process of Mentoring Program by assigning a staff member to be the contact person for the program. They can even assign their own staff member to be the Program Mentor.

  This type of collaboration between the company, mentor and mentee will likely to bring the tech skills gap closer.

User Stories - related to App Background Idea

Related to Company User

- *As an owner of a business, I would like to* avoid risky investment and only accept quality proposals that provide the solution to my project idea which has gone idle because of lack of manpower and proof of concept, *so that I* can bring my project into realisation.
- *As an owner of a business, I would like to* give a chance to a potential employee to work on a small project and get trained in a specific area, *so that I* could ensure the employee has the necessary skill sets to work on the similar or even bigger project within my company.
- *As an owner of a business, I would like to* give a chance to upskill talented people, *so that I* could help close the tech skills gap within the IT industry.
- *As an owner of a business, I would like to* give a chance to those within the minority groups who get disadvantaged because of their background, *so that I* could promote equal opportunity within the tech industry.

Related to Mentor User

- *As a professional, I would like to* give my time to mentor fellow/future IT employees, *so that I* could gain Professional Development hours.
- *As a professional, I would like to* work in a program that allows for collaboration with a company and a fellow/future IT employee, *so that I* could extend my network within the IT industry.
- *As a professional, I would like to* work as a mentor, *so that I* could get some extra income.
- *As a professional, I would like to* get a chance to work on more projects, *so that I* could improve my resume and get an opportunity for a job promotion.
- *As a professional, I would like to* see more companies giving a chance to people like me, who are considered as the minority groups, to get industry recognition by working in their small projects, *so that I* will have a chance to compete in the employment market.

Related to Mentee User

- *As a fresh graduate, I would like to* apply to a mentorship program, *so that I* could get some hands-on training in real-life projects and get the support from the experts.
- *As an employee, I would like to* apply to a mentorship program, *so that I* could be up-to-date with the current skills needed by the industry.
- *As a fresh graduate, I would like to* apply to a mentorship program, *so that I* could gain experience working in the IT industry.
- *As an employee, I would like to* apply to a mentorship program, *so that I* could get some extra income.
- *As an employee, I would like to* apply to a mentorship program, *so that I* could apply to a job with a better position.
- *As a fresh graduate, I would like to* see more companies giving a chance to people like me, who are considered as the minority groups, to get industry recognition by working in their small projects, *so that I* will have a chance to compete in the employment market.

Research on similar applications

There are few similar applications out there that offer a similar concept (mentoring program) but with different target audiences and end goal.
Some of these are:

- Score.org (2016b), which aims to help small businesses to establish and grow by offering a mentoring program and workshops. Its mentors also come from volunteers like the mentorship system. It also offers flexibility in terms of meeting approach which can be at Score local offices or remotely via phone, video or email. The collaboration is in the form of advice given by volunteers (Mentors) to the small businesses they are paired with. Small businesses do not have to pay for the service.

- FreeManagerMentors (2019a), which aims to help managers and leaders (including small businesses) in managing their human resource better, such as team communication, communicating with the supervisors, effective recruitment, etc.
- Kaleidoscope Initiative Australia (2020a), which aims to help new migrants who are already skilled workers (professionals) to get their skills acknowledged and gain employment in Australia. The initiative welcomes volunteers who are professionals from a variety of industries to come and be the mentor for a newcomer. In regard to cost, the website pointed out that a person who is interested to be a mentee will have to pay AUD $100.00 for when they have successfully matched to a mentor. It is not clear to whether or not there is an ongoing fee for the mentoring service.
- MercyCare Australia (2020b), which aims to help migrants to gain access to education, training and employment in Australia. It gives support giving advice and information on relevant education and training options, career planning, interview skills, as well as approaching companies to seek work opportunities.

Although the applications above have some similarities to BridgeInTech and support the community in their own way, they are different to BridgeInTech in the followings:

- They have a limited target on who can be considered to be qualified as a mentee. This poses a problem for people who are wanting support (to be a mentee) but are not within the category (for example, not a new migrant or a new migrant but with no work experience).
- With the exception of Scope.org, they offer support in a general area (such as human resource management, information on education, training and potential career), but not the actual hands-on learning program to further grow their skills in a particular area to enhance their career. Scope supports is more focused on the small businesses needs.
- To some extent, they have a weaker form of company affiliation as the effect of lack of incentive for the company to participate in the program. In BIT, one of the direct motivations for companies to be involved is a chance to unblock their idle project. Whereas the non-direct impact could be to see more skilled people in the industry and help close skills gaps

Technical Discussion

Since one of the purposes of BIT is to bring in industry involvement to the MS, I would integrate the MS into BIT. It would also be ideal if later in the future BIT also integrates AnitaB.Org Volunteer Management System (VMS) into the application. The reason why I choose the existing MS and VMS application instead of building from scratch is so that we avoid reinventing the wheel, to make project development faster, as well as to meet the timeline of GSoC.

In regard to MS integration, there are few options that I could think of on how to approach this:

1. Extend the current MS so as to include BIT entities and features (1FE + 1BE).
   Pros:
   - Data is kept in one place - MS DB
   - Less complicated in architecture design - using the current MS architecture
   - Less overhead (one server, one db)
   - Faster data retrieval (one source)
   - One GUI for the whole system (BIT + MS = White labelling)

   Cons:
   - Major changes to MS DB schema, such as adding a totally new entity (like Company and Program) and adding attributes to the existing entities, like User in MS will be extended to accommodate attributes for Personal_Background of BIT. Furthermore, Mentorship_Relation of MS will also need to be extended by adding attributes of Mentoring_Program_Relation of BIT.
   - Major addition to MS Android Frontend.
   - In short, causing significant disruption to MS development
2. Use the current MS as REST API and have separate Backend and Frontend specific to BIT (2FE + 2BE).
   Pros:
   - Minor change to MS DB schema. The changes needed are:
     - On User_Profile: adding Phone, Mobile, Personal_Website ('Required' features) and Resume ('Future' feature) (fig. 4.5)
   - Less disruption to MS development.

   Cons:
   - Data for the same Mentor/Mentee is kept in two different locations (User_Profile in MS with additional Personal_Background in BIT).
   - Place MS or BIT in dilema. Since the ideal way is for BIT to use MS Production API so that BIT development is less dependent on MS development. But this means to push MS to have small releases every

now then which will not be ideal to MS. On the other hand, if BIT is using MS Development API, it will disrupt BIT development as the constant code change/merge in MS might block certain BIT features.
    - More overhead (2 servers and 2 DBs)
    - Longer time for processing request (2 sources)
    - 2 separate GUIs (however, I like the idea of having a separate space (MS - GUI) to manage Mentor/Mentee relation as it would benefit mentor/mentee in a way that they can focus on tasks at hand)
3. The middle ground for the 2 options above is to use the current MS REST API but with 2 Frontend: MS and BIT. In short 2FE + 1BE.
   Pros:
    - Reduce overhead compared to option 2.
    - No need for MS Android FE to be changed to accommodate BIT UI
    - Being MS FE exclusive to Mentor/Mentee, their interaction throughout the program can be focused on the task at hand.
    - Minimise company interference on the specific details within the process of mentoring between mentor and mentee. Company involvement is limited to collaborating with mentor/mentee in designing tasks at the beginning prior to actual program starts. However, companies can still interact with the mentor and mentee through the IRC channel throughout the duration of the program.

   Cons:
    - Reducing company involvement on the tasks specific could be seen as unfavourable to some companies.

<u>Decision:</u> I will take the 3rd approach for integrating MS with BIT.

As for the tech stack, I will leave it to the AnitaB.org mentors and admins to decide which technology to be used. However, at the moment there are few models I could think of:

1. as per MS current tech stack which is:
   for Backend:
    - RESTful API with Python and Flask
    - Relational database either using mysql or postgresql
    - Remote development server hosted on Heroku

   for Frontend:
    - Android app with Kotlin

   Why? Because of my familiarity to the language and framework used through the current MS application.
2. as a Web application with Backend as above but Frontend using ReactJS.
   Why? I want to learn more about progressive app (Girish, 2017) because it has some ability to tap into native features of mobile app on top of being responsive.

3. as a Cross-platform application with Backend as above but FrontEnd using Flutter or React Native.
   Why? I want to learn more about the advantages of this type of framework as mentioned in an article by net solutions (Machanda, 2019) (code reusability, cost-effectiveness, consistency in UI components, easy hosting, cloud integration, fewer technical barriers, and shorter time to market). I'm more inclined to learn in Flutter because being The "new kid on the block", it has already been reported to be within the top 3 most loved framework that makes it sounds cool so I want to learn more.

When I started gsoc journey, my initial idea was to have the app as the web application because of the responsiveness to different media sizes and no cost (no need to place it on the app store). Which is why I started my mockups with the web app in mind. However, I got inspired by Bartek Pacia's presentation on Flutter on one of the zoom project feedback meeting sessions and started thinking about a different approach to the tech stack (using Flutter).
To summarise, my first preference to tech stack is option 3, followed by 2 with the reason already explained above. However, due to my limited experience, with the two frameworks, it is understandably that option 1 would be my best option.
Again, I'm flexible and a fast learner so would adapt to the decision made between admins and mentors on how to move forward on this project.

Other Challenges

On top of the challenges in the form of technical decisions to be made as per discussion in the previous section, there are other challenges that I have to be mindful of. These challenges are:

1. Demographic data and analysis.

   Although demographic analysis will be put aside and looked at in a later future (beyond GSoC), there is still the current Personal_Background data of mentor and mentee that needs to be collected as part of the required feature of MVP. The concerns in regard to collecting this data are:
   - what options should be put in the dropdown selector for each category on the 'Personal Background' on Create_Profile page (figure 4.6) so as to be inclusive and covers all different types of group that exist in each category?
   - how to make sure that the language/term being used is acceptable to address the specific type of group in a category without being seen as offensive?
   - how 'willing' the users would associate themself to be in a specific category? What is the unseen impact to the users when they do this

(like community acceptance - family, work, friends - where they belong)? How to help lift this mental barrier?

In an ideal world, everyone should help each other to build a positive and inclusive community. However, reality tells us we still have a long way to go. This statement is supported by OECD National Health Surveys from 10 countries (2015) which shows people with mild-to-moderate mental illness are twice likely to be unemployed compared to those without. This situation still exists regardless of the fact the issue was raised over decades ago (Clay, 2012). In the ladder of acceptance towards the minority group for employment, mental illness is the least favourable factor for the company in looking for a potential employee (O'Hara, 2007; Baldwin and Marcus, 2007), I assume this is more apparent in a highly demanding and stressful type of job like IT. I raise this concern because I fear that including 'Mental Ability' as one of the categories within the minority group in the app, will do more harm than good to people who are in this category. This was also the reason why I did not include 'Mental Ability' in the first place. If we do not have it as a category of the minority groups in the app, this will not be the barrier that will limit their chances to get selected in a BIT Mentoring Program or to be considered as a potential employee beyond BIT program.

Nevertheless, I could see myself struggling whether or not to answer such a question if I were to be in the user's shoes. Such a mental barrier I believe is not only unique to me.

<u>Decision:</u> My personal preference is to not include 'Mental Ability' as one of the categories of the minority group (~ Personal_Background table; Target Candidate in Create_Program page) as I fear the negative effect it will bring to the people this application is trying to support. However, I will leave the final decision in AnitaB.org's hand.

2. <u>Database type</u>

The decision on the initial database type that will be adopted in the project will somewhat affect the demographic analysis or any other future analysis that later could be developed as additional features. More research needs to be done to make sure the database is scalable and suitable to the project needs beyond GSoC period to accommodate this 'Future' features. Things to think about:

- Single vs multiple db. Which one better?
- If multiple, which one to go for? SQL + SQL vs SQL + NoSQL?
- Which one is more suitable for data analysis? Or even ML?

<u>Decision:</u>

With the consideration of the existing MS DB + REST API and following the suggestions from the mentors (Ramit Sawhney and Monal Shadi), I have opted to use the 3rd option on the Technical Discussion section. In this approach, I will use Relational DB SQL (PostgreSQL) with two schemas (MS

and BIT schemas) in MS DB while using only MS REST API as a server with API responses in JSON format.
This structure is chosen because single DB makes it easier for schema management and performance tuning, easier to build reports, as well as high availability and disaster recovery (Ozar, 2011). Beside, Flask-SQLAlchemy made it possible to have multiple schemas to talk to each other and be in a relationship (2016a). Furthermore, the Python+PostgreSQL combo in this structure is ML ready thanks to Python ML Libraries such as Kmeans clustering (Resnisky, 2018). However ML will not be included within the GSoC20 development. This can be added later beyond GSoC.

3. Timeline and project scope

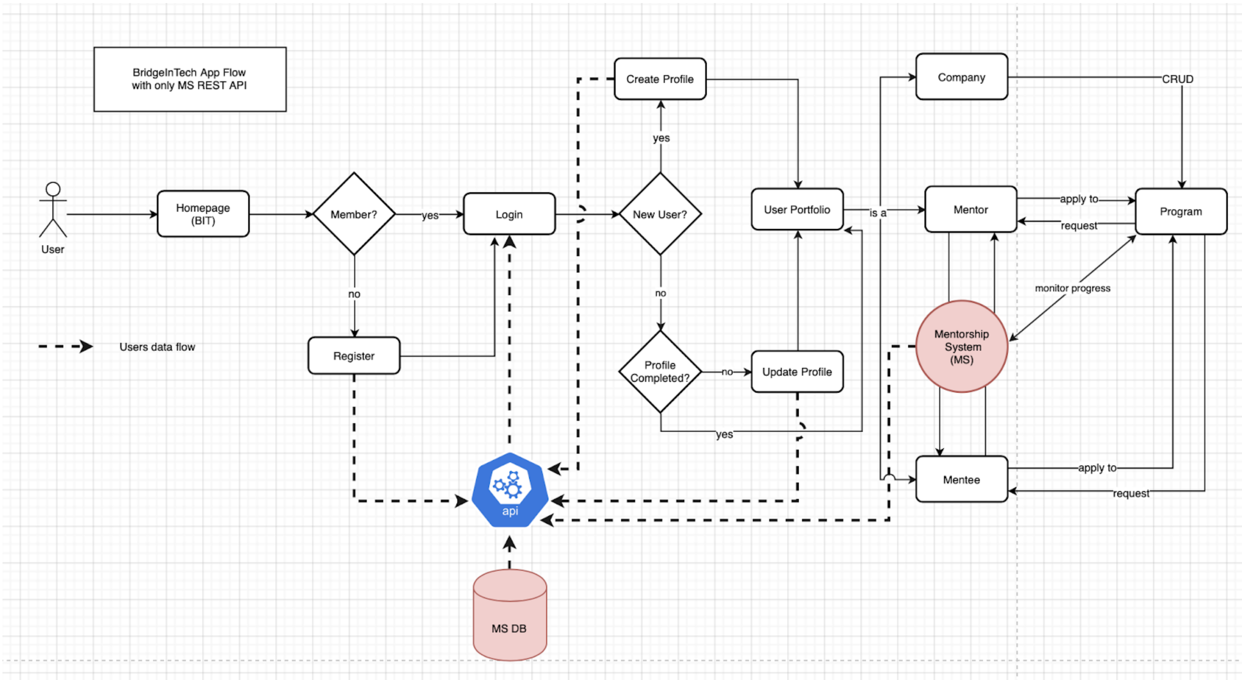
As with all projects, the target MVP needs to be matched with the timeline of the project. Having to develop both BIT backend and frontend as well as trying to integrate them with the existing MS project, coupled with being a newcomer to the industry, I have had concerns when I set the timeline. My approach of the timeline is an optimist approach. I will do my best to complete the goal set in the timeline.
Previous experience:
I have had some experience about Python and Flask through the online courses offered on Testdriven.io
Here are the repositories of the projects (note that access to these repositories is currently set to 'Private' to respect the copyright and the intention of the course - being a paid course - from the author. Access to these repositories will be given upon request only):
=> related to Python, Flask, Flask-RESTPlus API, Flask-SQLAlchemy, Flask-Admin, Postgres, Pytest, TDD testing, Docker, Gitlab CI and Swagger. More information about tools and technologies used can be found here (2019d).
The repository I created for my learning:
https://gitlab.com/mytravelpa01/flask-tdd-docker.git

=> related to everything learnt from the course above plus Authentication using JWTs, Flask-Bcrypt, Flask-CORS, React, Formik, Testing with Jest and React Testing Library. More information about tools and technologies used can be found here (2019c).
The repository I created for my learning:
https://gitlab.com/mytravelpa01/flask-react-auth.git

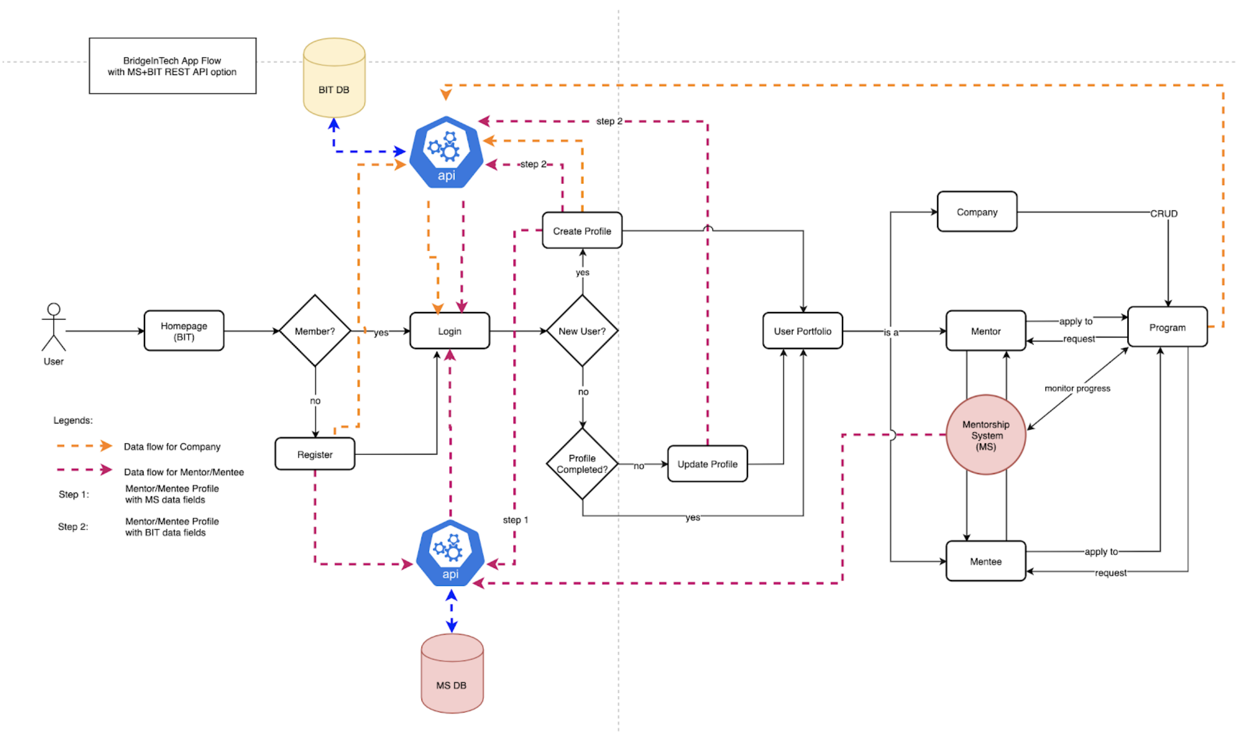
Application Flow

Link to online document: https://drive.google.com/open?id=1Cxa3l449Odpb3XMeghrH1W01I7ScVLTV

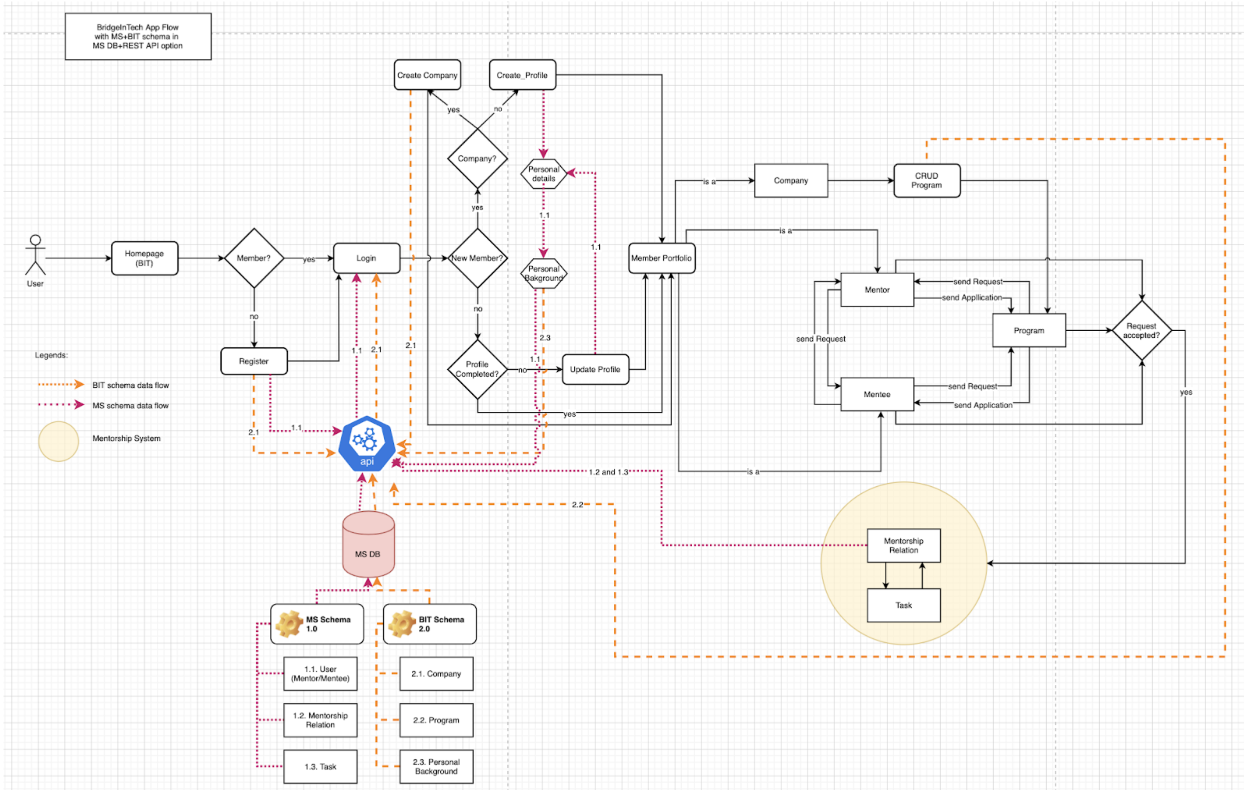
Opt. 1. With MS REST API + DB Only



Opt. 2. With MS + BIT REST APIs + DBs

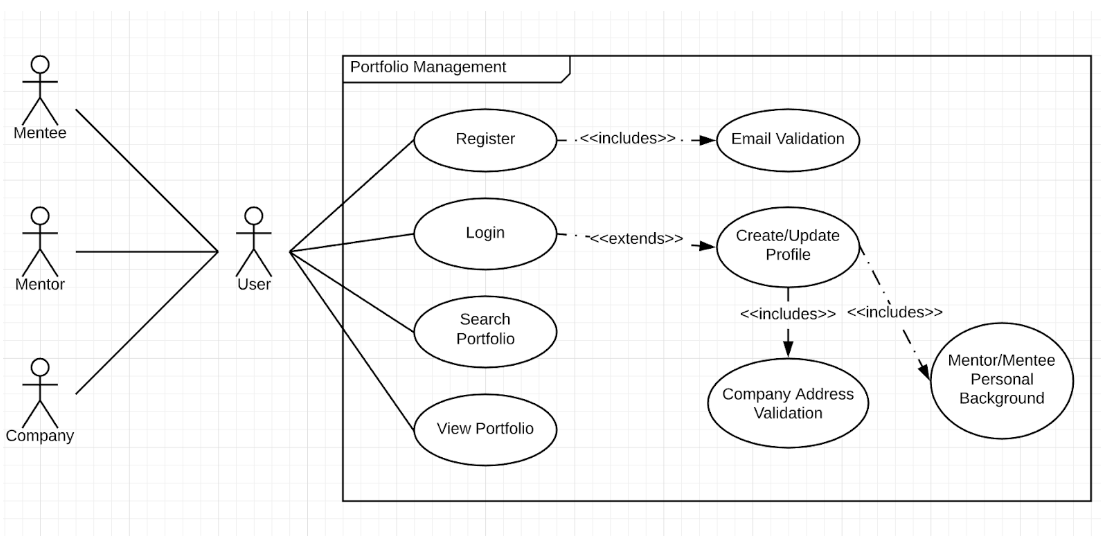


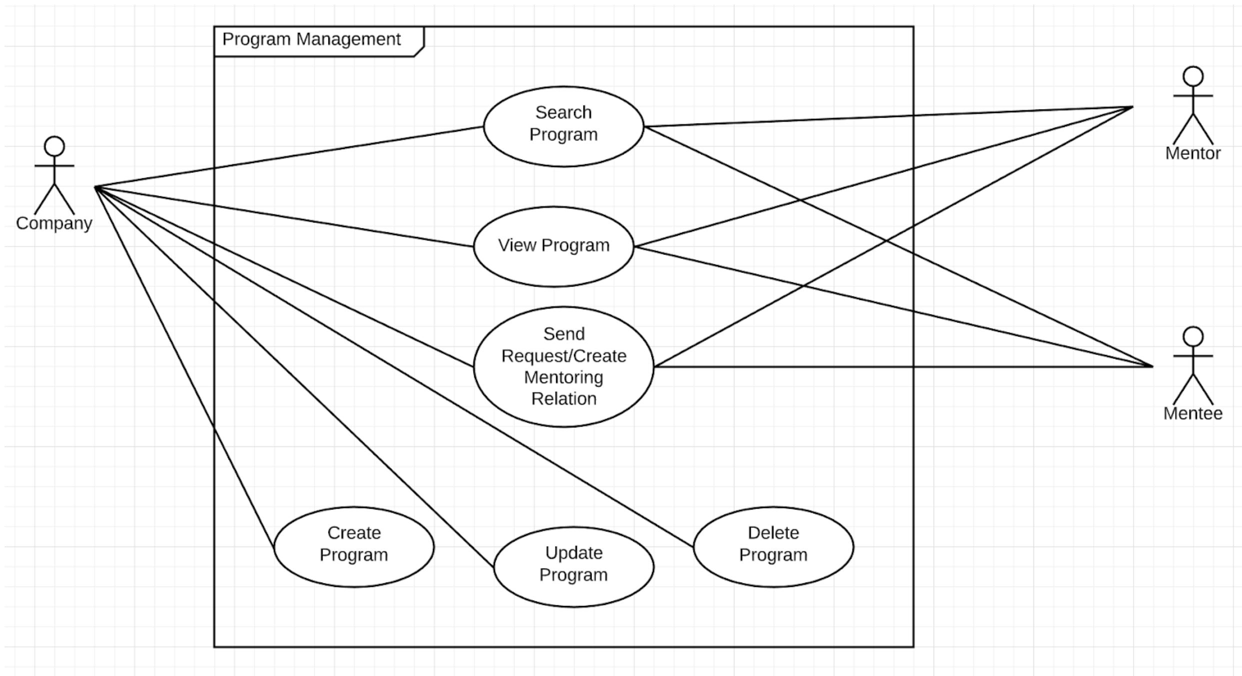
Opt. 3. MS+BIT schemas in MS DB+REST

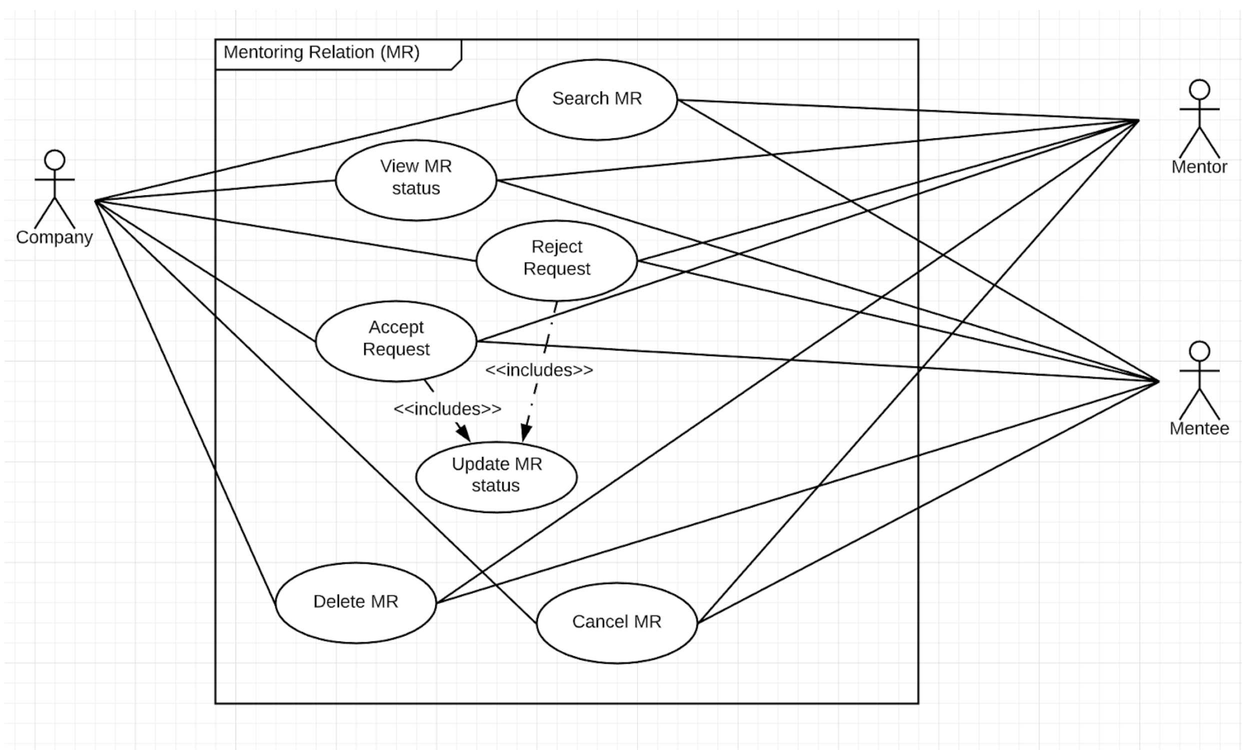


Use Cases

Link to online document: https://www.lucidchart.com/invitations/accept/bc04ae11-0953-41bb-902a-4abb58ea4efa




Program Management
Company
Search Program
View Program
Send Request/Create Mentoring Relation
Create Program
Update Program
Delete Program
Mentor
Mentee


Mentoring Relation (MR)
Company
Search MR
View MR status
Reject Request
Accept Request
<<includes>>
<<includes>>
Update MR status
Delete MR
Cancel MR
Mentor
Mentee

User Stories - related to Application Features

On General Features - MVP

- *As a public and member, I would like to* see a public landing page which would have information about what BridgeInTech is about, a snapshot of the recent Mentoring Programs they have, and how to contact them if I have questions, filtering and sorting ability on the recent programs listed, and clickable links that will direct me to get more information on: what the BridgeInTech is about, a specific company, a specific mentor/mentee, a specific program, *so that I* can use it as a starting point when I want to use the application and decide whether I want to participate in any of the programs being offered, contact BridgeInTech, filter and sort through the list, and go to the page referred to by the clickable links. (Feature: Homepage and its content - Fig.1.3 MockUp)
- *As a public and member, I would like to* see a consistent navbar consist of tabs to Home, Programs, Companies, Mentors, Mentees, Register and Login (or Logout for logged in user) and search fields, across all public pages of the application, *so that I* can get familiar quickly, browse faster and find my way around the application through clicking the tabs. (Feature: Public Navbar - across all Public pages)
- *As a public and member user, I would like to* see the list of all companies in BridgeInTech which shows their name, skills they offer to mentor in, their location and timezone, the number of programs they have, the number of programs with target candidate, and the status of their programs, a clickable link to a specific company portfolio, and a filtering and sorting abilities of each columns shown in the list, *so that I* can decide whether there is a particular company I would like to work with in the Mentoring Program, filter and sort through the list, and go to the page referred to by the clickable links. (Feature: Companies page and its content - Fig. 1.4 MockUp)
- *As a public and member user, I would like to* see the list of all programs offered in BridgeInTech which shows the programs name, which company offers them, the skills they are targeting, whether or not they have target candidate, their current status (open/in progress/closed), start and end dates, the mentor and mentee involved in them, their current progress, clickable links to: a specific program details, a specific company portfolio, a specific mentor or mentee portfolio, and a filtering and sorting abilities of each columns shown in the list, *so that I* can decide whether or not to apply for any particular program being offered, filter and sort through the list, and go to the page referred to by the clickable links. (Feature: Programs page and its content - Fig. 2.1 MockUp)
- *As a public and member user, I would like to* see the list of mentors and mentees which shows their name, the skills they have, the number of

programs they have completed, their availability status, their location and timezone, and their email address, clickable link to a specific mentor/mentee portfolio, and a filtering and sorting abilities of each columns shown in the list, *so that I* can decide whether I would like to request the mentor or mentee to work with me in a program, filter and sort through the list, and go to the page referred to by the clickable link. (Feature: Mentors/Mentees page and its content - Fig. 2.2 MockUp)

- *As a public and member user, I would like to* see the portfolio of the specific company that I'm interested to work with, which would have a brief information on who they are, list of programs they have, their contact details, and a clickable link to: the program details *so that I* can decide whether the company will be the right one for me to be involved with in a mentoring program, and go to the specific program details page. (Feature: Company Portfolio and its content - Fig. 2.3 MockUp)
- *As a public and member user, I would like to* see the information details of a mentoring program which would have a description of what the program is about, the company who offers it, the start and end dates, the current status, the name of mentor/mentee or whether the position is still available, target skills that will be develop in the program, tags on range of tools and skills involved, the target candidates, whether it is a paid or non-paid program, the contact type, contact details and the responsibilities of mentor, mentee and the company involved, a clickable link to the company portfolio, buttons whether to send an application or to check progress (if this is a program I'm currently involved in), *so that I* can decide whether I want to apply to the program if it is still available, go to the company portfolio page, apply to the program (or to see program progress if I'm currently involved in it), or just to have an idea of the trend in the industry in tech skills needed from the past or current program. (Feature: Program Details page and its content - Fig. 3.1 MockUp)
- *As a public and member user, I would like to* see the portfolio of a mentor or a mentee that I'm interested to work with, which would have a brief information on who they are, list of programs they had or currently involved in, their contact details, buttons to send mentor/mentee request to them (if I'm a mentor/mentee user) or send company request (if I'm a company user), and clickable links to the following pages: the mentor/mentee profile, their personal background (if made public), program details, company portfolio, mentor/mentee portfolio this member is working with, *so that I* can get relevant information about this member to see if they would be the right fit for me to work with in a mentoring program, send request for the member to work with me in a program, and to go to the page referred to by the clickable links. (Feature: Mentors/Mentees Portfolio page and its content - Fig. 3.2 MockUp)

- *As a public user, I would like to* be able to become a member of BridgeInTech, *so that I* can use the features offered by them through their application. (Feature: Register - Fig.1.1 MockUp)
- *As a member user, I would like to* have an area within the application that is dedicated to me either as a mentor/mentee or company user, *so that I* can use it to keep track of my activities, showcase my portfolio and programs I had or currently involved in (for mentor/mentee) or have to offer (for company user), and apply to/get offer to work in (for a mentor/mentee user) a mentoring program; or create a mentoring program and receive applications from/send offer to mentor/mentee interested to work in the program (for company user). (Feature: Login - Fig.1.2 MockUp)

On Specific Features - MVP

- *As a logged in user, I would like to* be able to go to my portfolio page, *so that I* can see a recap of my activities in BridgeInTech. (Feature: Company Profile for company user - Fig. 2.3 MockUp or Mentor/Mentee portfolio for Mentor/Mentee user Fig. 3.2 MockUp)
- *As a logged in user, I would like to* see a consistent navbar consist of tabs to Home, Portfolio, Events, Blogs and Logout and search fields, across all pages where login is required, *so that I* can get familiar quickly, browse faster and find my way around the application through clicking the tabs. (Feature: Logged in members Navbar - across all pages where login is required)
- *As a company user, I would like to* be able to create a company profile which would have: the company name, brief description about my company, contact details containing address, phone number, email, website, as well as contact person details who will represent my company, *so that I* can have the contact information on my company portfolio page for people to contact my company when they are interested about a program I offer. (Feature: Create Company Profile - Fig. 3.3 MockUp)
- *As a company user, I would like to* have a dedicated page to create a program which would contain information on: a section for program details with the program name, my company name, program start and end dates, description, target skills, drop down selector for target candidate categories with options limited to the current minority groups within the tech industry, payment currency selector and payment type field for when I’m offering a program with a stipend, contact types; a section on the student, mentor and company responsibilities; a section on the student and mentor requirements as well as what resources my company is going to provide; a section for the program contact details which by default auto-filled with my company contact details to which I can change if the contact details of the program is different to the one in my company profile. This program contact details will have the contact person’s name, the department they belong to in my company, the

address of where the program is going to be held, contact person's phone number and/or mobile, program email address, website and the IRC channel Url where the communication for the program will be mainly held throughout its duration. Finally, the program should have a program status section with the default value of 'Draft' and can be modified manually to 'Open'. This Create_Program functionality is *so that I* can post and announce the program my company has to offer to the BridgeInTech community. (Feature: Create Program - Fig. 4.1 - 4.3 MockUp)

- *As a company user, I would like to* send a request to a mentor or a mentee which would contain the following information: the program name, the program id number, my company name, the program url, the name of mentor or mentee I send the request to, their user id, their full name, location, phone number, the position offered to them, and a message, *so that I* can request the mentor or mentee i'm interested in to work in my mentoring program. (Feature: Company Request Form and submission - Fig. 4.4 MockUp)
- *As a mentor or mentee user, I would like to* send a request to a company on a program it offers which would contain the following information: my username, my user id, my full name, location, phone number, program name, program id number, company name, the position I'm applying for, and a message, *so that I* can request the company which program i'm interested in to take me in to work on the mentoring program. (Feature: Application Form and submission - Fig. 4.5 MockUp)
- *As a mentor or mentee user, I would like to* create a profile which would contain the following information: a section for personal details with username, IRC ID, the type of availability (available to mentor/need mentoring), my interests, bio, location, occupation, skills, email, as well as a section for personal background with options to be selected for gender, age, ethnicity, sexual orientation, religion, physical and mental ability, socio economic, highest education, and years of experience. I can choose to have the information on personal background to be kept private and not to disclose information to the BridgeInTech community and general public. I would like to have this Create_Profile page, *so that I* can post my information so that I can be contacted by the mentor/mentee/company who would be interested to work with me on a program. (Feature: Create Profile - Fig. 4.6 MockUp)

MockUps

Minimum Viable Product (MVP)

On General Features

- Homepage
- Register and login
- Company/Mentor/Mentee Portfolio
- See list of all Companies
- See list of all Programs
- See list of all Mentors
- See list of all Mentees
- See Program Details page

On Specific Features

- Company can create a Company Profile and do other CRUD operations on it.
- Mentor/Mentee can create Mentor/Mentee Profile and do other CRUD operations on it.
- Company can create a Program and do CRUD operations on Program Details
- Company can send/accept request to/from Mentor/Mentee to work on a Program
- Mentor/Mentee can send/accept request to/from Company to work on a Program
- Mentor/Mentee relationship will be managed in **MS**
- Users can see Mentorship Progress Summary (link to tasks completion in **MS**)

List of MockUps

Related to All Users

- Register (figure 1.1)
- Login (figure 1.2)
- Homepage (figure 1.3)
- Companies (list) (figure 1.4)

- Programs (list) (figure 2.1)
- Mentors (list) *(Mentees is not shown as one of the MockUps because it has similar layout to Mentors)* (figure 2.2)
- Company Portfolio (figure 2.3)
- Program Details (figure 3.1)
- Mentor Portfolio (figure 3.2)

Related to Company User

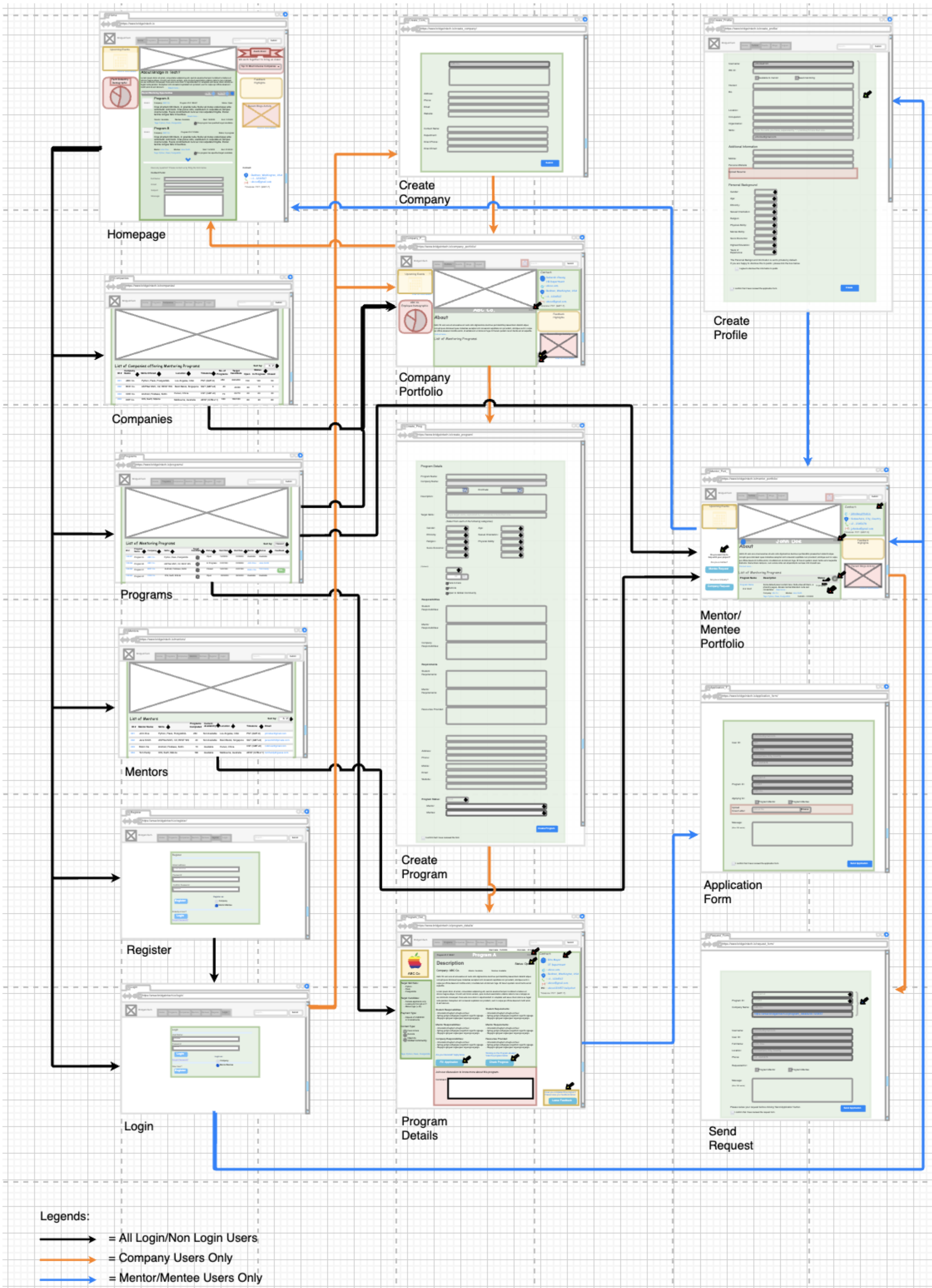
- Create Company Profile (figure 3.3)
- Create Program (figure 4.1 - 4.3)
- Send Request (figure 4.4)

Related to Mentor/Mentee User

- Application Form (figure 4.5)
- Create Profile (figure 4.6)

Feature Flows

Link to online document: https://drive.google.com/open?id=1jw13JVScwWSnesw6LnRqR0ynn_09E884



MockUp Screenshots

Link to online document: https://drive.google.com/open?id=1oiT6Ez2IqNKcBT9Y3nkFeVfq7FHVFUWD

Notes for MockUps:

- CRUD functionalities will be only available to the user who is the owner of the information on the page. E.g:
    - On Company_Portfolio, the 'edit' and '+' only visible/accessible to Company user.
    - On Mentor_Portfolio, the 'edit' and '+' only visible/accessible to a Mentor user.
- on colour scheme:
    - Green highlighted area indicates the 'Required' features as stated in the Project Features.
    - Yellow highlighted area indicates the 'Nice to have' features.
    - Red highlighted area indicates the 'Future' features.
    - The form with only black/grey and white indicates it is also 'Required' features (except if a certain area is highlighted in red - 'Future' feature).
    - Red arrows (on Company_Portfolio, Program_Details and Mentor_Portfolio) indicate the functionalities that will be hidden to non-login users.
    - Yellow arrows (on Company_Portfolio, Program_Details and Mentor_Portfolio) indicate that button visibility will depend on the Program 'status'.
    - Green arrows:
        - On Mentor_Portfolio, 'Mentor Request' button links to 'Send_Request' in MS; the 'Company_Request' button links to 'Request_Form' in BIT; 'Profile' links to User_Profile in MS; and 'Background' links to Personal_Background details in BIT.
        - On Create_Profile MockUp indicates fields that will be filled with existing data taken from MS DB if the new Login user is a new Mentor/Mentee user to BIT but an existing user of MS.
    - Blue arrow on Company_Request_Form indicates the fields where options in the selector will be taken from existing data in BIT DB. 'Program URL' will turn grey and auto filled with data once Program ID is selected.

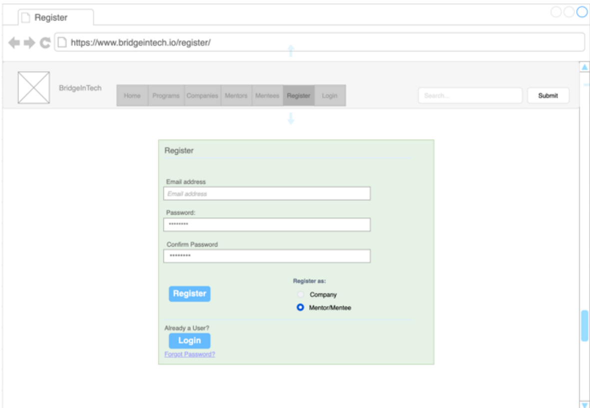


Figure 1.1. Register

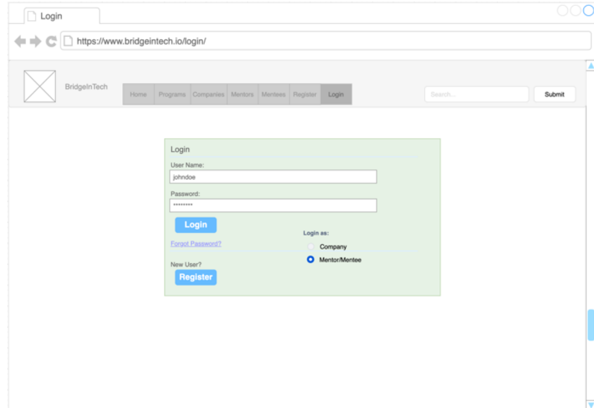


Figure 1.2. Login

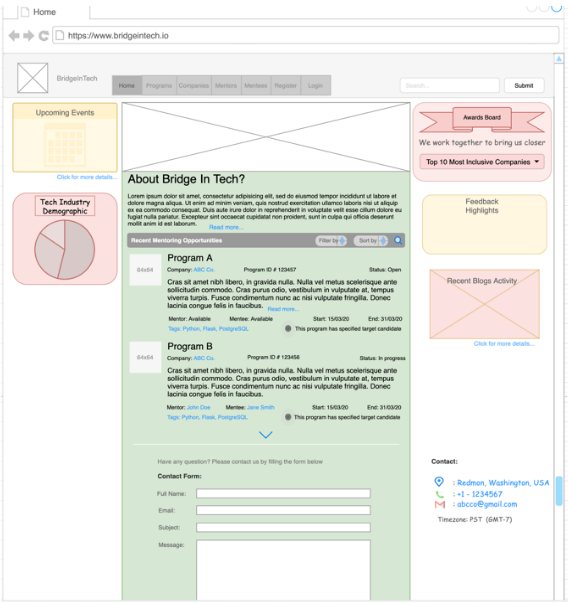


Figure 1.3. Homepage

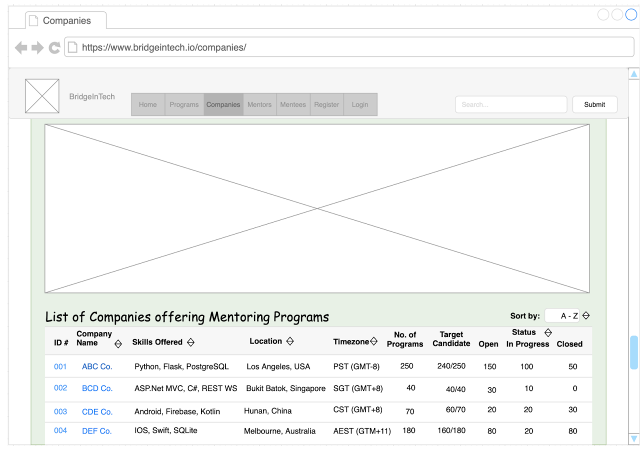


Figure 1.4. Companies/list

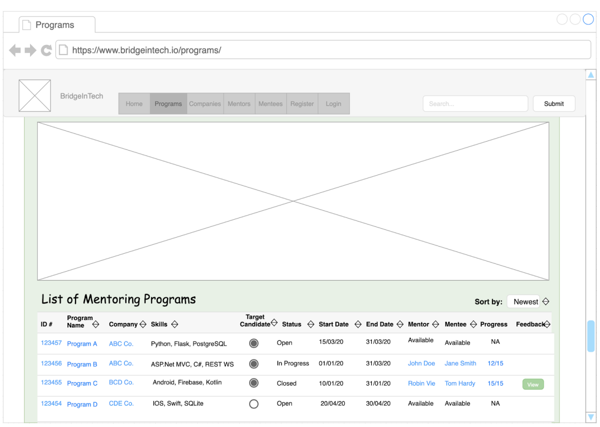


Figure 2.1. Programs/list

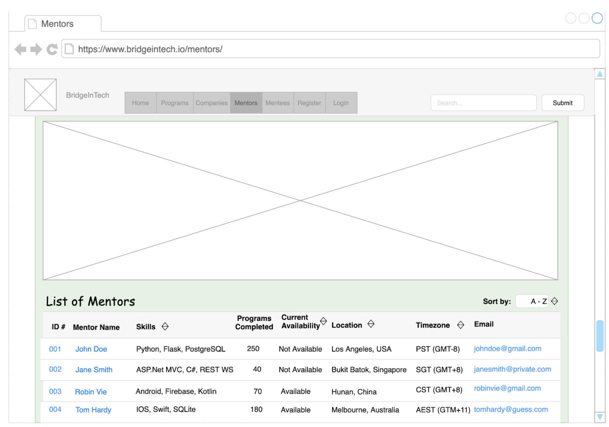


Figure 2.2 Mentors/Mentees list

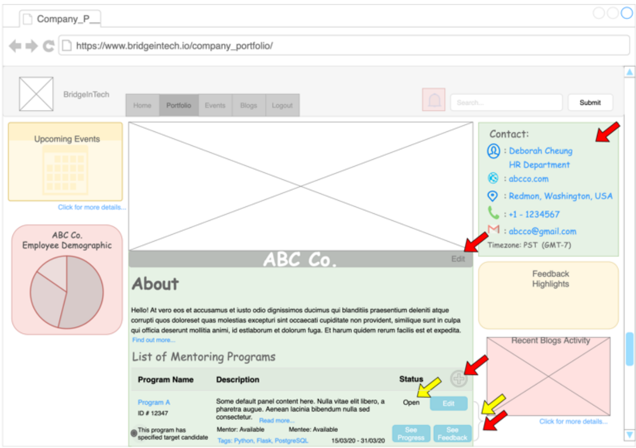


Figure 2.3. Company Portfolio

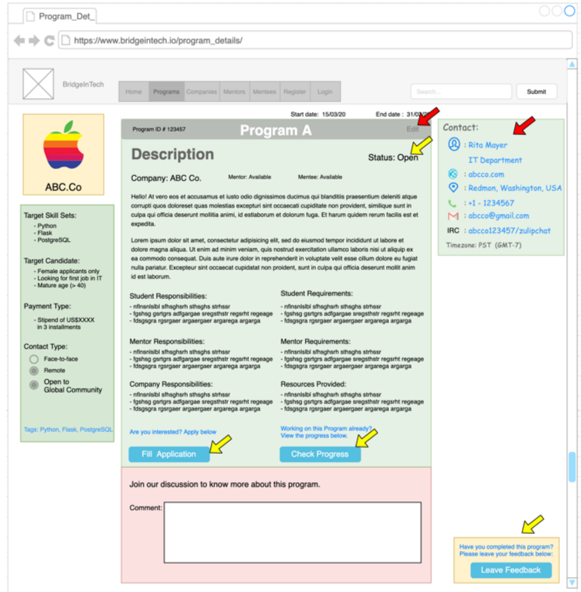


Figure 3.1. Program Details

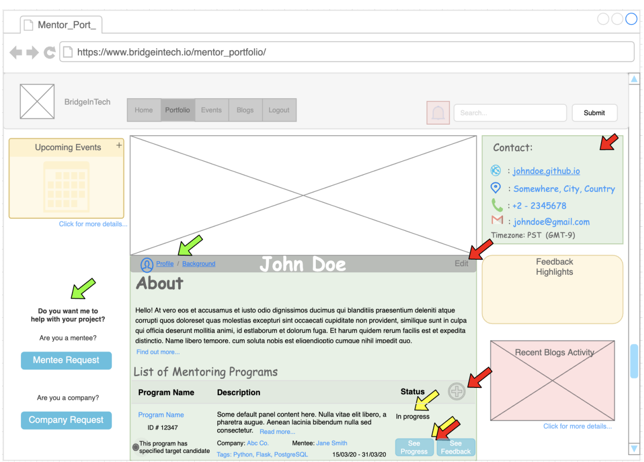


Figure 3.2. Mentor Portfolio

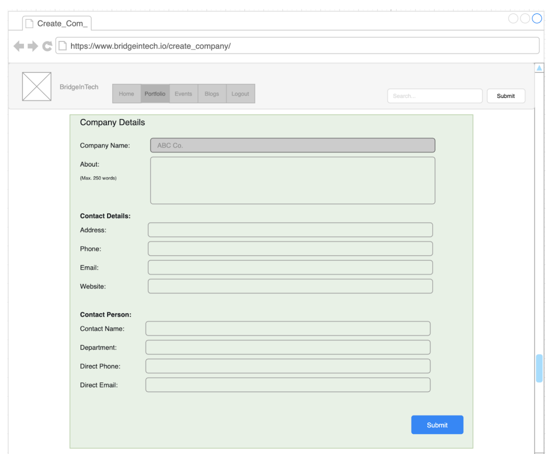


Figure 3.3. Create Company Profile

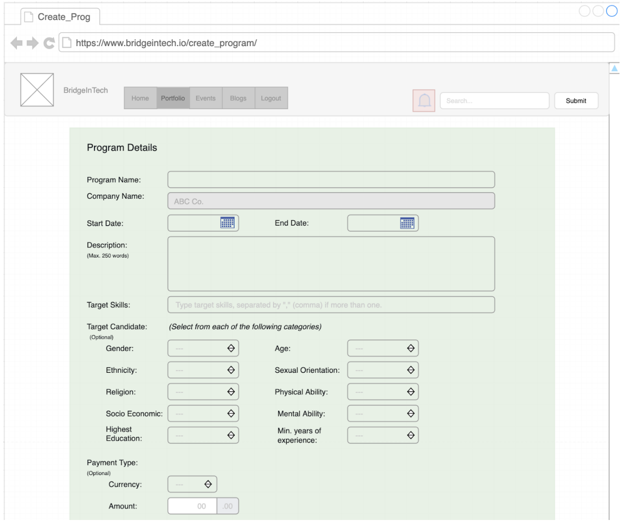


Figure 4.1. Create Program (1of3)

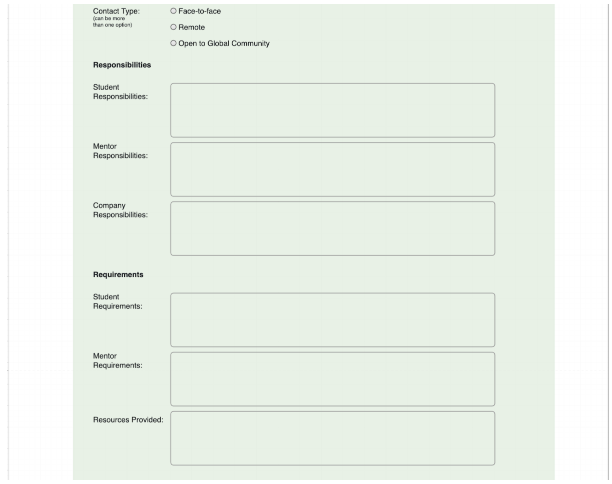


Figure 4.2. Create Program (2of3)

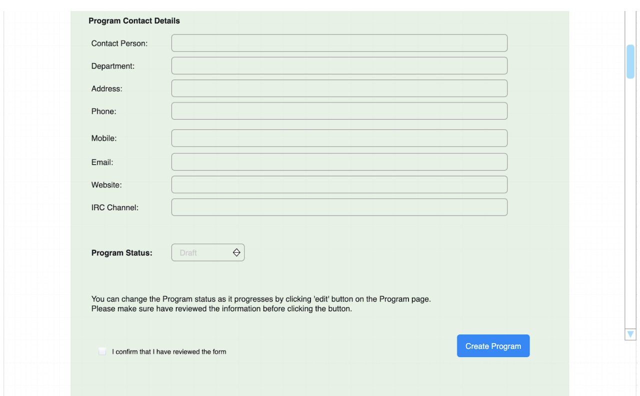


Figure 4.3. Create Program (3of3)

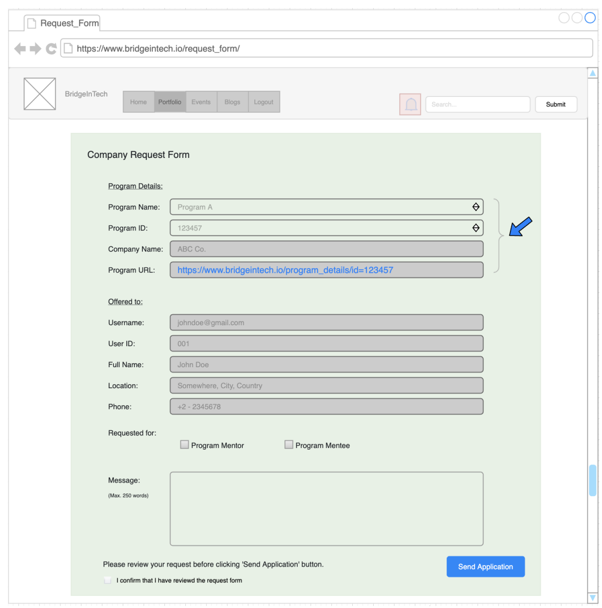


Figure 4.4. Company Request Form

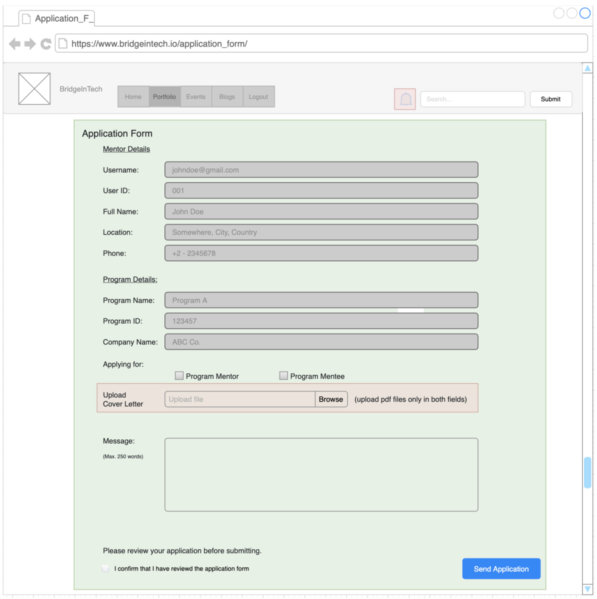


Figure 4.5. Application Form

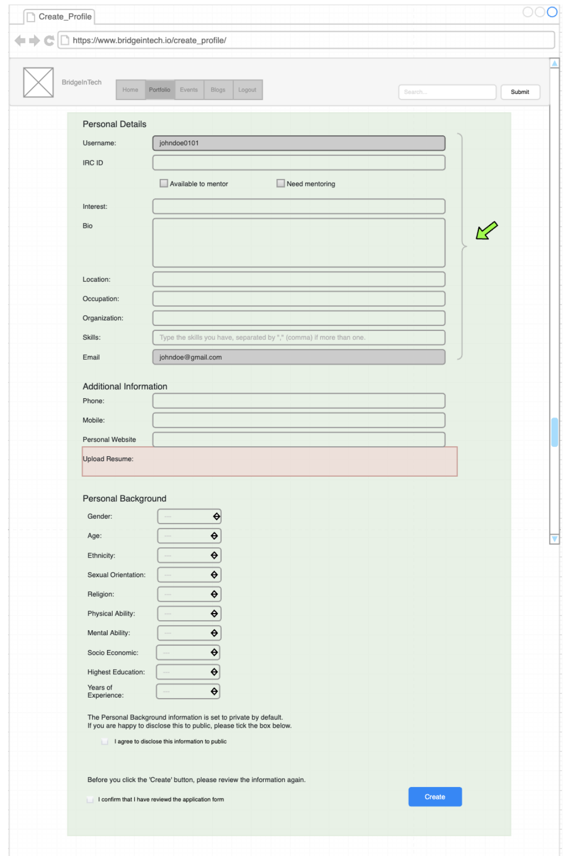


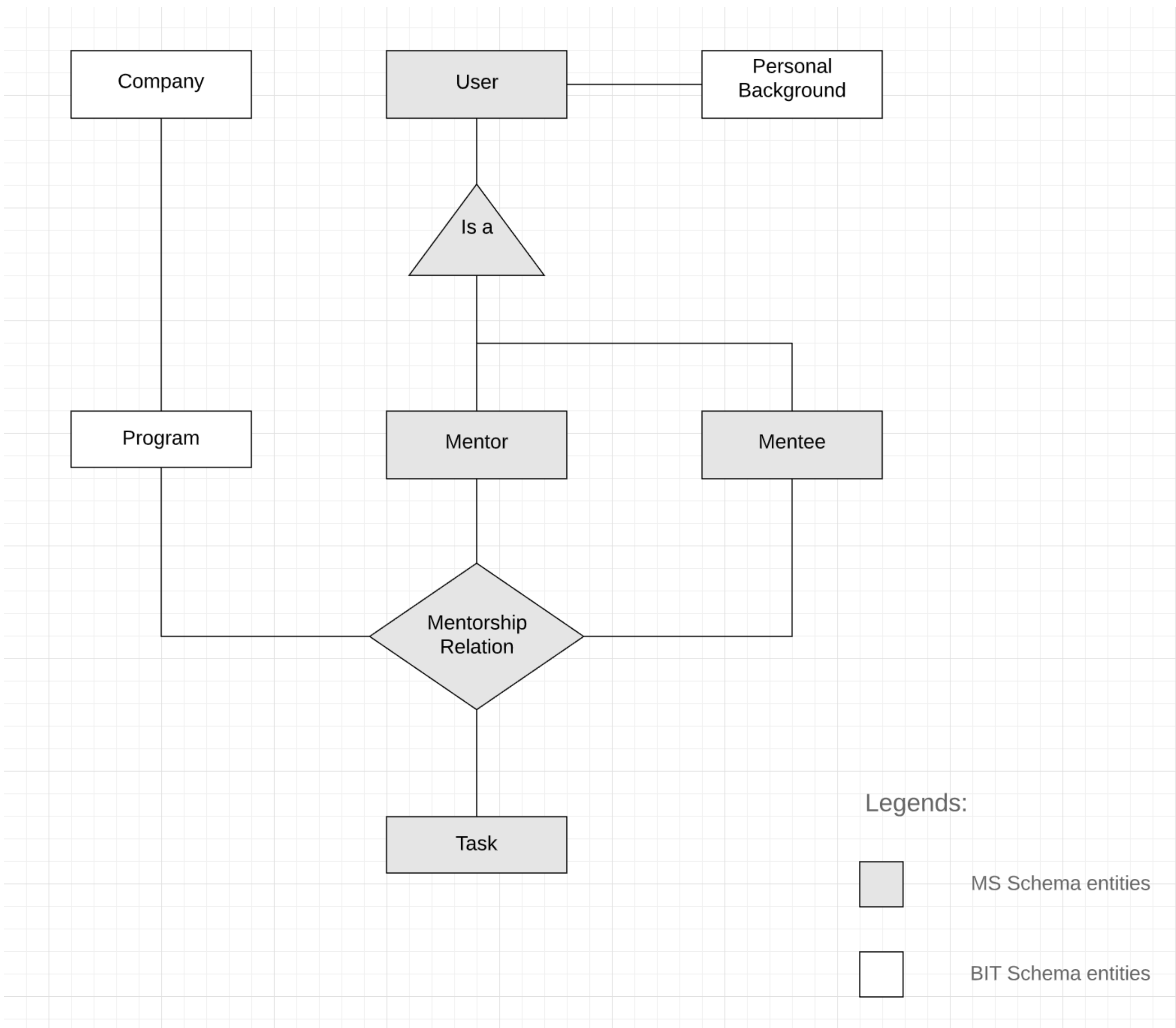
Figure 4.6. Create_Profile

System Design Decisions

Link to online document (Entity Association Model and Entity Relationship Diagram):
https://www.lucidchart.com/invitations/accept/123e1020-1ebb-4eb3-9dea-1b101f22b357

Note: in this section, Mentorship_Relation will be referred to as **MR**

Entity-Association Model



Relationship Constraints

- Users can only register once. This includes a Company user which can only have one account.
- A Program cannot exist without a Company.
- Company is in One-to-Many relation to Program.
- Personal_Bakground and Additional_Info cannot exist without User (**MS**)
- Personal_Background and Additional_Info are in One-to-One relation to User (**MS**)
- There can only be One-to-One relation between Company, Mentor and Mentee in the **MR**.
- Mentor/Mentee can only have one **MR** active at a time.

Data Normalisation Analysis

(Under the assumption Mentoring relation will be managed in one entity without looking at potential disruption to MS/BIT development.)

MENTORING_RELATION

UNF

MENTORING_RELATION (mentoring_relation_id, program_id, program name, program_description, initiation_date, start_date, end_date, is_completed, mentor_id, mentor_name, mentor_accept_date, mentee_id, mentee_name, mentee_accept_date, message, creation_date, action_id, state, notes, tasks_list)

1NF

MENTORING_RELATION (<u>mentoring_relation_id</u>, <u>program_id</u>, <u>mentor_id</u>, <u>mentee_id</u>, program name, program_description, action_id, state, mentor_name, mentor_accept_date, mentee_name, mentee_accept_date, message, creation_date, initiation_date, start_date, end_date, is_completed, notes, tasks_list)

mentoring_relation_id, program_id, mentor_id, mentee_id **->** program name, program_description, action_id, state, mentor_name, mentor_accept_date, mentee_name, mentee_accept_date, message, creation_date, initiation_date, start_date, end_date, is_completed, notes, tasks_list **FULL**

program_id **->** program_name, program_description, start_date, end_date **PARTIAL**

mentor_id **->** mentor_name **PARTIAL**

mentee_id **->** mentee_name **PARTIAL**

<u>2NF</u>

MENTORING_RELATION (<u>mentoring_relation_id</u>, <u>program_id</u>, <u>mentor_id</u>, <u>mentee_id</u>, action_id, state, mentor_accept_date, mentee_accept_date, message, creation_date, initiation_date, is_completed, notes, tasks_list)

PROGRAM (<u>program_id</u>, program_name, program_description, start_date, end_date)

MENTOR (<u>mentor_id</u>, mentor_name)

MENTEE (<u>mentee_id</u>, mentee_name)

<u>3NF</u>

MENTORING_RELATION (<u>mentoring_relation_id</u>, <u>program_id</u>, <u>mentor_id</u>, <u>mentee_id</u>, action_id, state, mentor_accept_date, mentee_accept_date, message, creation_date, initiation_date, is_completed, notes, tasks_list)
PROGRAM (<u>program_id</u>, program_name, program_description, start_date, end_date)
MENTOR (<u>mentor_id</u>, mentor_name)
MENTEE (<u>mentee_id</u>, mentee_name)

COMPANY

<u>UNF</u>

COMPANY (company_id, name, about, c_address, c_phone, c_email, c_website, c_contact, c_department, c_direct_phone, c_direct_email, (program_id, program_name, start_date, end_date, description, target_skills, target_candidate, payment_currency, payment_amount, contact_type, student_responsibility, student_requirements, mentor_requirements, resources_provided, p_contact_name, p_department, p_address, p_phone, mobile, p_email, p_website, irc_channel, status))

1NF

COMPANY (company_id, company_name, about, c_address, c_phone, c_email, c_website, c_contact, c_department, c_direct_phone, c_direct_email)

company_id -> company_name, about, c_address, c_phone, c_email, c_website, c_contact, c_department, c_direct_phone, c_direct_email **FULL**

PROGRAM (program_id, company_id, program_name, start_date, end_date, description, target_skills, target_candidate, payment_currency, payment_amount, contact_type, student_responsibility, student_requirements, mentor_requirements, resources_provided, p_contact_name, p_department, p_address, p_phone, mobile, p_email, p_website, irc_channel, status)

program_id, company_id -> program_name, start_date, end_date, description, target_skills, target_candidate, payment_currency, payment_amount, contact_type, student_responsibility, student_requirements, mentor_requirements, resources_provided, p_contact_name, p_department, p_address, p_phone, mobile, p_email, p_website, irc_channel, status **FULL**

2NF

COMPANY (company_id, company_name, about, c_address, c_phone, c_email, c_website, c_contact, c_department, c_direct_phone, c_direct_email)
PROGRAM (program_id, company_id, program_name, start_date, end_date, description, target_skills, target_candidate, payment_currency, payment_amount, contact_type, student_responsibility, student_requirements, mentor_requirements,

resources_provided, p_contact_name, p_department, p_address, p_phone, mobile, p_email, p_website, irc_channel, status)

3NF

COMPANY (company_id, company_name, about, c_address, c_phone, c_email, c_website, c_contact, c_department, c_direct_phone, c_direct_email)
PROGRAM (program_id, company_id, program_name, start_date, end_date, description, target_skills, target_candidate, payment_currency, payment_amount, contact_type, student_responsibility, student_requirements, mentor_requirements, resources_provided, p_contact_name, p_department, p_address, p_phone, mobile, p_email, p_website, irc_channel, status)

Collected 3NF Relations

1. MENTORING_RELATION (mentoring_relation_id, program_id, mentor_id, mentee_id, action_id, state, mentor_accept_date, mentee_accept_date, message, creation_date, initiation_date, is_completed, notes, tasks_list)
2. PROGRAM (program_id, program_name, program_description, start_date, end_date)
3. MENTOR (mentor_id, mentor_name)
4. MENTEE (mentee_id, mentee_name)
5. COMPANY (company_id, company_name, about, c_address, c_phone, c_email, c_website, c_contact, c_department, c_direct_phone, c_direct_email)
6. PROGRAM (program_id, company_id, program_name, start_date, end_date, description, target_skills, target_candidate, payment_currency, payment_amount, contact_type, student_responsibility, student_requirements, mentor_requirements, resources_provided, p_contact_name, p_department, p_address, p_phone, mobile, p_email, p_website, irc_channel, status)

Attribute Synthesis

3 & 4 => MENTOR == MENTEE aka USER

USER (<u>user_id</u>, user_name)

Additional attributes of USER from BIT collected on Create_Profile page:
phone, mobile, website, gender, age, ethnicity, sexual_orientation, religion, physical_ability, mental_ability, socio_economic, highest_education, years_of_experience

Additional attributes of USER from MS:
username, password_hash, email, terms_and_conditions_checked, registration_date, is_admin, is_email_verified, email_verification_date, current_mentorship_role, membership_status, bio, location, occupation, organization, slack_username, social_media_links, skills, interests, resume_url, photo_url, need_mentoring, available_to_mentor

So together, BIT and MS have the following attributes of USER:
USER (<u>user_id</u>, user_name, username, password_hash, email, terms_and_conditions_checked, registration_date, is_admin, is_email_verified, email_verification_date, current_mentorship_role, membership_status, bio, location, occupation, organization, slack_username, social_media_links, skills, interests, resume_url, photo_url, need_mentoring, available_to_mentor, phone, mobile, website, gender, age, ethnicity, sexual_orientation, religion, physical_ability, mental_ability, socio_economic, highest_education, years_of_experience)

COMPANY (<u>company_id</u>, company_name, about, c_address, c_phone, c_email, c_website, c_contact, c_department, c_direct_phone, c_direct_email)

PROGRAM (<u>program_id</u>, <u>company_id</u>, program_name, start_date, end_date, description, target_skills, target_candidate, payment_currency, payment_amount, contact_type, student_responsibility, student_requirements, mentor_requirements, resources_provided, p_contact_name, p_department, p_address, p_phone, mobile, p_email, p_website, irc_channel, status)

MENTORSHIP_RELATION (<u>mentorship_relation_id</u>, <u>program_id</u>, <u>mentor_id</u>, <u>mentee_id</u>, action_id, state, mentor_accept_date, mentee_accept_date, message, creation_date, initiation_date, is_completed, notes, tasks_list)

(Note that the start_date and end_date of Mentorship_Relation is no longer needed in this table as it is part of the Program table. The same goes with accept_date which is replaced by mentor_accept_date, mentee_accept_date)

Schema joins vs Postgres Json

This analysis is to decide whether to add the additional attributes for Mentoring_Relation or user additional info/personal background as a separate schema in BIT that joins with their relevant entities in MS (Mentoring_Program_Relation to Mentorship_Relation; Additional_Info and Personal_Background to User in MS schema) upon queried, or to include the attributes of these entities inside their correspondent entities in MS as Postgres json column.

The decision to choose between either schemas or postgres jsonb column instead of EAV was due to the following findings in my research:

EAV vs Json -> answer: Json. why? Because Json has some advantages over EAV: Less lines of codes; much simpler query; faster performance and smaller table size. Although using json makes the code a bit more complex and not as easy to read compared to EAV (Coussement, 2016).
Json vs Jsonb -> what and which to use? Answer: Jsonb. Why? Json doesn't support indexing (Halliday, L 2015; Markou, 2018).

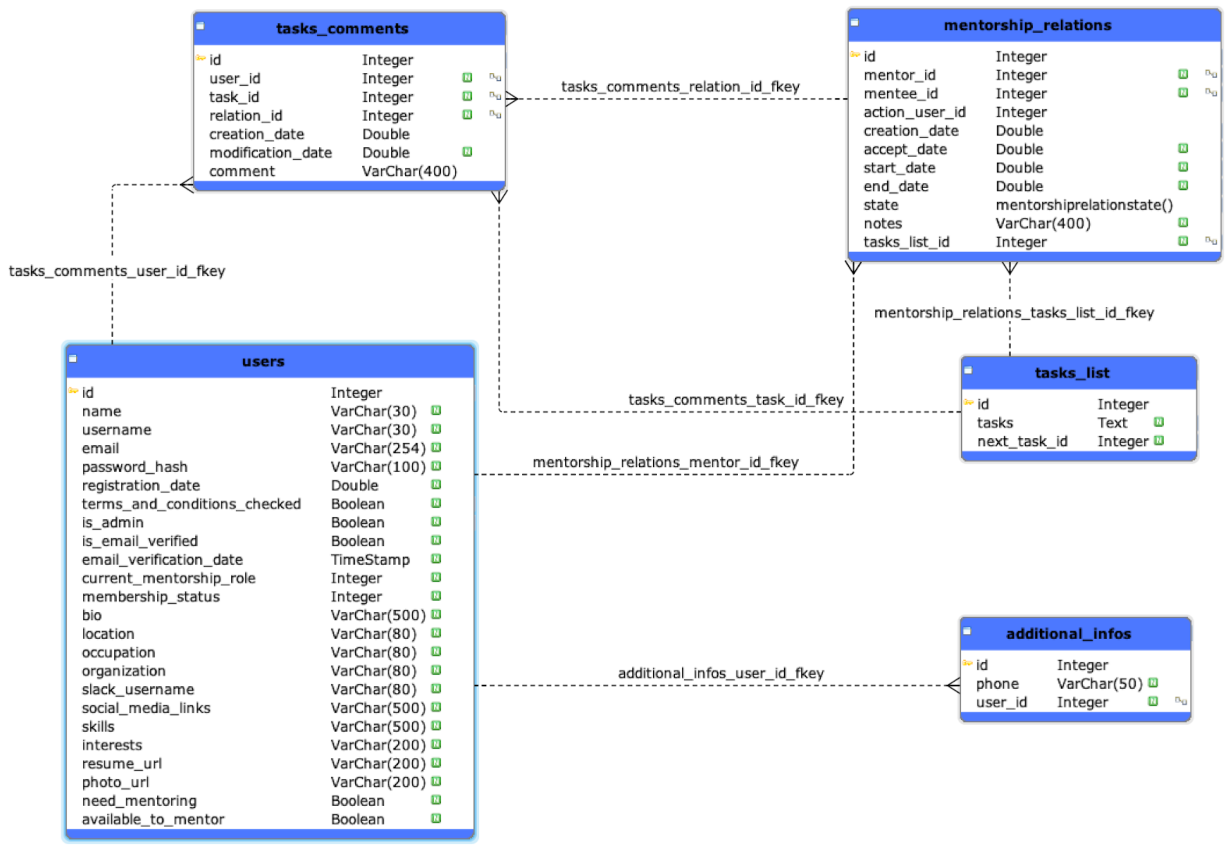
Now, the analysis…

Existing Mentorship System ERD:

Scenario 1: As separate schemas

The test was done on the existing mentorship-system backend code on a separate branch from the develop branch in my forked repository. I added a separate schema into MS DB (called bitschema) and created an entity Additional_Info with attributes phone and user_id (an FK from USER MS).

Test ERD on dual MS+BIT schemas:

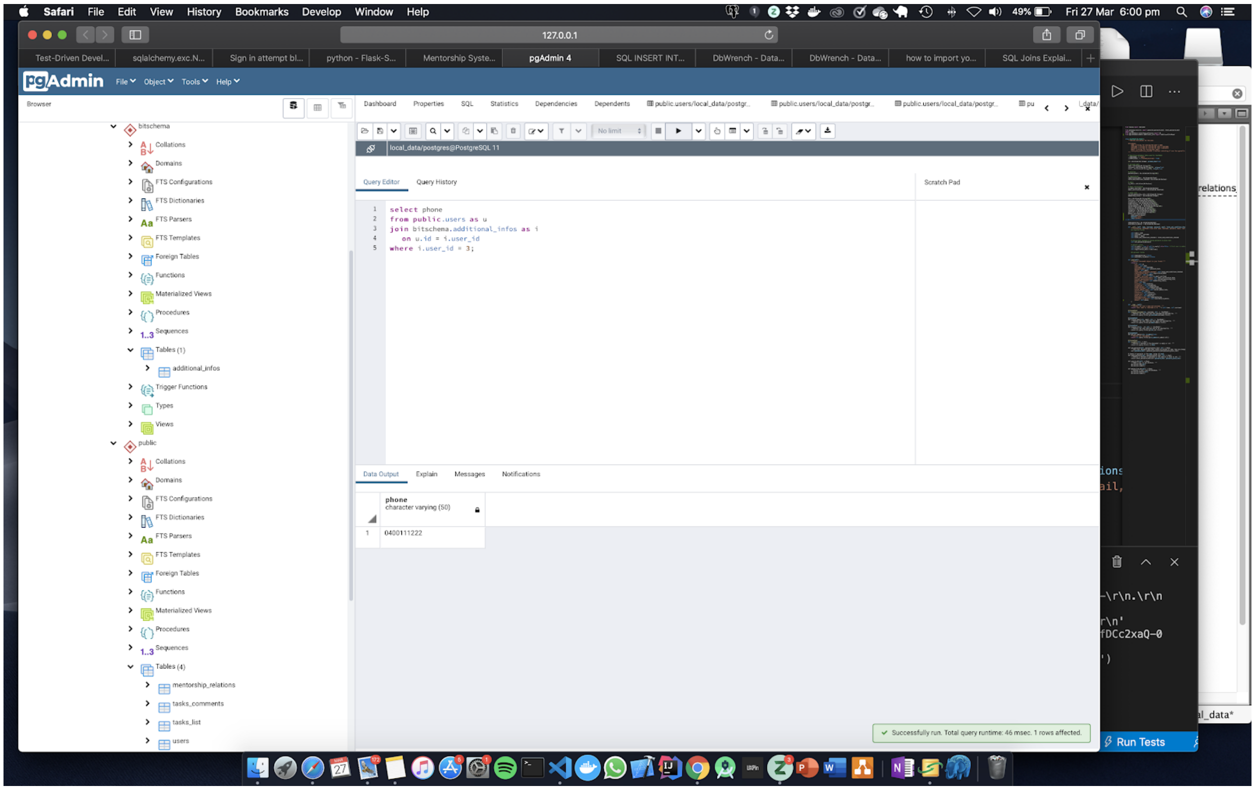


The entity User (with auto created id = 3) was populated with the following data using Swagger UI:

User
{
  "name": "Test",
  "username": "test0101",
  "password": "test0101",
  "email": "mtreacy@mail.com",
  "terms_and_conditions_checked": true,
  "need_mentoring": true,
  "available_to_mentor": true
}
Additional_Info was populated with the following data using pgadmin4 with SQL query:

{
  "phone": "0400111222,

"User_id": "3"
}
Then a query was run on [pgAdmin4](#) to get the phone number of user 3, the result was given within 46 msec.



Code changes made to UserModel:

user.py .../database/... additional_info.py .../models

app > database > models > user.py > UserModel

```
    current_mentorship_role = db.Column(db.Integer)
    membership_status = db.Column(db.Integer)

    bio = db.Column(db.String(500))
    location = db.Column(db.String(80))
    occupation = db.Column(db.String(80))
    organization = db.Column(db.String(80))
    slack_username = db.Column(db.String(80))
    social_media_links = db.Column(db.String(500))
    skills = db.Column(db.String(500))
    interests = db.Column(db.String(200))
    resume_url = db.Column(db.String(200))
    photo_url = db.Column(db.String(200))
    phone = db.relationship(
        AdditionalInfoModel,
        backref="owner",
        lazy="dynamic",
    )
```

New Entity created AdditionalInfoModel:

```python
from app.database.sqlalchemy_extension import db

class AdditionalInfoModel(db.Model):
    __tablename__ = "additional_infos"
    __table_args__ = {"schema": 'bitschema'}

    id = db.Column(db.Integer, primary_key=True)
    phone = db.Column(db.String(50))
    user_id = db.Column(db.Integer, db.ForeignKey("users.id"), unique=True)

    def __init__(self, phone, user):
        self.phone = phone,
        self.user_id = user,

    def json(self):
        """Returns information of mentorship as a json object."""
        return {
            "id": self.id,
            "phone": self.phone,
            "user_id": self.user_id,
        }

```

Pros:

- Simpler codes and easier to query, which leads to shorter processing time.

Cons:

- Data is stored in two different locations and must use joins to be queried. This is in particular not ideal for User entity

Scenario 2: As Postgres jsonb column in MS

With the readily found data in the research materials laid at the beginning of this Schema vs Postgres Json analysis, I will not reinvent the wheel trying to run my own test by following the same steps as I have done in scenario 1 above. Instead, I will present the views stated in the following materials: Halliday (2015), Markou (2018),

Pros:

- More suitable for the arbitrary data.
- More human-readable.
- More suitable for data that is siloed or isolated from other data.

- Support indexes and data is stored in binary format (~ no parsing needed).
- Support containment and existence test.

Cons:

The following disadvantages of using Jsonb is stated in Robinson (2016) and Ringer (2015):
- Slow queries due to lack of statistics which cause production issues. Later down the track when we start doing analysis, this will be a problem.
- Larger overhead in terms of disk space.
- Compared to normal relational database queries, jsonb code looks less readable.
- No fixed data typing which means higher chance for bugs especially in the areas where we are dependent on user's input (such as phone numbers, skills type, etc).
- No constraints. A normal CHECK, ENUM or DOMAIN sql operation must be done through writing own methods in the application.

I'll keep these points in mind when I get to the point of making the decision on which design suits this project best.

Discussion:
- Without calculating the impact on overhead (disk space) and considering that the user data in the ADDITIONAL_INFO table is more of a one directional query (from USER to ADDITIONAL_INFO table, reverse is not true) and does not have any analytic value, jsonb is more suitable to be used as the data type for ADDITIONAL_INFO attributes.
  One directional query example:
  -> USER to ADDITIONAL_INFO: what is the user phone number?
  Why is the reverse not true?
  -> ADDITIONAL_INFO to USER: Who's phone number is this?
  (Unlikely we'll ever asked such a question to query the database)

- Since we do need to make some analysis on Demographic Data of the users later down the track as part of the 'Future' feature of the application, it is best to use the relational database with separate schema for user's personal background data (in PERSONAL_BACKGROUND table). The same goes with COMPANY and PROGRAM where both have data that will be used in further analysis in the 'Future' future (demographic data, analysis of types of skills, candidates, contact type, etc). The data on these tables are also bidirectional which means querying will be much faster to be done to schema joins compared to extracting jsonb. Plus, less overhead used in running the query. MENTORSHIP_RELATION will also have all attributes as relational data type because of its extensive joins with other tables and bidirectional query potentials.

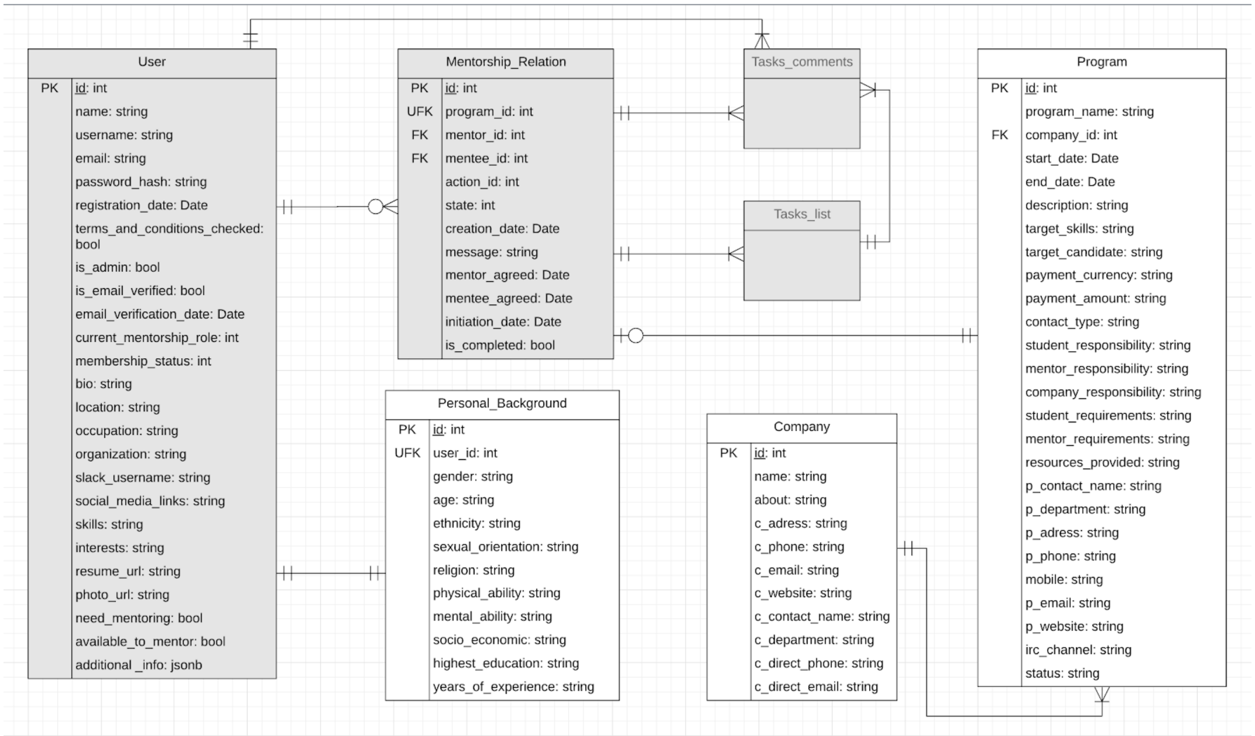
- Before I decide to use jsonb in USER table as alternative to have a separate ADDITIONAL_INFO table (relational data), I will consider the following:
    - (-) code writing and querying of this data type will be different to the current flow of all other tables.
    - (-) increase complexity in code writing since the data needs to be deserialized/serialized and encoding/decoding for it to be used and mixed with other relational data (Python, 2020).
    - (+) using a separate entity for the attributes in this table might be redundant since most fields potentially be empty (beside for the compulsory phone number, the user might opt to not put their mobile or personal website details)

<u>Decision:</u>

Based on the data normalization and schemas vs json analysis discussed above, I will apply the following to the BIT project:

- To merge additional attributes of Mentorship_Relation both of BIT and MS into one table. This can be done by extending the existing MR(MENTORSHIP_RELATION) of MS.
- To apply relational models to all tables other than ADDITIONAL_INFO. The attributes of the ADDITIONAL_INFO will be added as jsonb in the existing USER table (MS).
- To keep PERSONAL_BACKGROUND table as a separate table because the need for analysis later in the future.

Entity Relationship Diagram



Note: Grey tables belong to MS schema and white tables belong to BIT schema

Notes:

- The User table in **MS** is to store data on Mentor/Mentee whereas for company user, their data is stored in the Company table.
- Program can have its own address, phone, mobile, email, website that are different to those of Company. For example when Company HQ is in California, USA but the Program is held in its Florida branch.

Table relationship explains

This section only explains tables that will be created in the BIT schema or the tables of MS that are affected by the project (Mentorship_Relation and User). The Tasks_list and Tasks_comments are not going to be explained.

MENTORSHIP_RELATION (<u>mentorship_relation_id</u>, program_id, mentor_id, mentee_id, action_id, state, mentor_accept_date, mentee_accept_date, message, creation_date, initiation_date, is_completed, notes, tasks_list)

program_id: FK (Program)
mentor_id, mentee_id: FK (User - **MS**)
Unique value: program_id
Default value: is_completed = False
mentor_id != mentee_id
Mentor and Mentee can only have/be in one Mentordhip_Relation at a time.

COMPANY (company_id, company_name, about, c_address, c_phone, c_email, c_website, c_contact, c_department, c_direct_phone, c_direct_emai)
Nullable fields: c_direct_phone, c_direct_email
Character limits: 3 < email < 320; 8 < phone < 50; 3 < website < 2047; 1 < contact_name < 100; 1 < department < 100; 8 < direct_phone < 50; 3 < direct_email < 320
Company must have a valid address.

PROGRAM (program_id, company_id, program_name, start_date, end_date, description, target_skills, target_candidate, payment_currency, payment_amount, contact_type, student_responsibility, student_requirements, mentor_requirements, resources_provided, p_contact_name, p_department, p_address, p_phone, mobile, p_email, p_website, irc_channel, status)
company_id: FK (Company)
Nullable fields: target_candidate, payment_currency, payment_amount, mobile
Default values: p_contact_name = c_contact_name; p_department = c_department; p_address = c_address; p_phone = c_phone; p_email = c_email; p_website = c_website; status: Draft
Minimum period: 4 weeks; Maximum period: 6 months.
Program must have a valid address.

USER (user_id, user_name, username, password_hash, email, terms_and_conditions_checked, registration_date, is_admin, is_email_verified, email_verification_date, current_mentorship_role, membership_status, bio, location, occupation, organization, slack_username, social_media_links, skills, interests, resume_url, photo_url, need_mentoring, available_to_mentor, additional_info)
Unique values: username, email
Nullable fields: occupation, organization, social_media_links, mobile, website

Character limit: 1 < social_media_links < 500; 1 < bio < 500; 1 < occupation < 50; 1 < location < 100; 1 < slack_username < 100; 1 < skills < 300; 1 < interests < 300;
additional_info: a jsonb object with data from key-value: {‘phone’: ‘ ‘, ‘mobile’: ‘ ‘, ‘website’: ‘ ‘}

PERSONAL_BACKGROUND (id, user_id, gender, age, ethnicity, sexual_orientation, religion, physical_ability, mental_ability, socio_economic, highest_education, years_of_experience)
user_id: FK (User - **MS**)
Unique value: user_id
Default value: Prefer not to say (on all attributes except PK and FK)

Notes:

- Set a trigger to determine the **MR** initiation_date. These conditions must be met for **MR** to be initiated: mentorship_relation_id != null; mentor_id != null; mentee_id != null; mentor_agreed != null; mentee_agreed != null. Once these conditions are met, the initiation_date of the **MR** is set to the system date of when the conditions are fulfilled.
- A Company can send a request (Request_Form - figure 4.4) to a Mentor or Mentee to work on a Program. When a company sends the request, company_id is the action_id of the **MR.**
- A Mentor or Mentee can send a request (Application_Form - figure 4.5) to a Company to work on a Program. When they do this, their user_id becomes the action_id of the **MR**.
- External Map API such as OpenStreetMap will be used to get the exact location of a company (~ address validation). This will be useful for querying purposes (e.g. show a list of Companies from a specific country)
- Program status can be modified manually from ‘Draft’ to ‘Open’ by the company user when they are ready to offer the program to the BridgeInTech community. The status of ‘In-Progress’, ‘Completed’ will be automatically updated by the application through a trigger to determine if the system date is within or past the program date. The Program status will only be modified to ‘Open’ if the MR related to the program has an initiation date on it (not null).

Higher Level API (CRUD operations)

Entity	**Operations**
User	Basic - create_user(name, username, password, email, terms_and_conditions_checked, need_mentoring, available_to_mentor) - getUser(user_id) - update_user(user_id, name, username, bio, location, occupation, organization, slack_username, social_media_links, skills, interests, resume_url, photo_url, need_mentoring, available_to_mentor, phone, mobile, website) - delete_user(user_id) - get _users() Extended - get_users([available_to_mentor, location) - get_users([need_mentoring, location]) - Etc… Or, to simplify the get methods above: - get_users(criteria) Where 'criteria' is a map of key-value (~dictionaries)
Personal_ Background	Basic - add_personal_background(user_id, gender, age, ethnicity, sexual_orientation, religion, physical_ability, mental_ability, socio_economic, highest_education, years_of_experience) - get_user_background(user_id) - update_personal_background(user_id, gender, age, ethnicity, sexual_orientation, religion, physical_ability, mental_ability, socio_economic, highest_education, years_of_experience) - delete_personal_background(user_id) Extended - list_target_users(physical_ability) - list_target_users(gender) - list_target_users([gender, age, years_of_experience]) - etc... Or, to simplify the list methods above: - list_target_users(criteria)

	Where ‘criteria’ is a map of key-value (~dictionaries)
Company	<u>Basic</u> - create_company(name, about, c_address, c_phone, c_email, c_website, c_contact_name, c_department, c_direct_phone, c_direct_email) - get_company(company_id) - update_company(company_id) - delete_company(company_id) - list_companies() <u>Extended</u> - get_country(company_id) - list_companies(country) - get_state(company_id) - list_companies([country, state]) - get_city(company_id) - list_companies([country, state, city]) - get_suburb(company_id) - list_companies([country, state, city, suburb]) - etc... Or, to simplify the list methods above: - list_companies(criteria) Where ‘criteria’ is a map of key-value (~dictionaries)
Program	<u>Basic</u> - create_program(program_name, company_name, start_date, end_date, description, target_skills, target_candidate, payment_currency, payment_amount, contact_type, student_responsibility, mentor_responsibility, company_responsibility, student_requirements, mentor_requirements, resources_provided, p_contact_name, p_department, p_address, p_phone, mobile, p_email, p_website, irc_channel, status) - get_program(program_id) - update_program(program_id) - delete_program(program_id) - list_programs() <u>Extended</u> - list_programs(contact_type) - list_programs(p_address) - list_programs(status)

	- list_programs(target_candidate) - list_programs(target_skills) - list_programs(payment_currency) - list_programs([p_address,status, payment_currency]) - etc... Or, to simplify the methods above: - list_programs(criteria) Where ‘criteria’ is a map of key-value (~dictionaries)
Mentorship_Relation	Basic - create_mentorship_relation(program_id, mentor_id, mentee_id, action_id, state, mentor_accept_date, mentee_accept_date, message, creation_date, initiation_date, is_completed) - list_mentorship_relations() - find_mentorship_relation_by_id(mentorship_relation_id) - accept_mentorship_relation_request(user_id, request_id) - reject_mentorship_relation_request(user_id, request_id) - cancel_mentorship_relation(user_id, request_id) - delete_mentorship_relation(user_id, request_id) Extended - list_current_mentorship_relations(user_id) *(all state)* - list_pending_mentorship_relations(user_id) - list_past_mentorship_relations(user_id) - list_rejected_mentorship_relations(user_id) - list_current_mentorship_relations(program_id) *(all state)* - list_pending_mentorship_relations(program_id) - list_rejected_mentorship_relations(program_id) - list_current_mentorship_relations(company_id) *(all state)* - list_pending_mentorship_relations(company_id) - list_rejected_mentorship_relations(company_id) - etc... Or, to simplify the list methods above: - list_mentorship_relations(criteria) where ‘criteria’ is a map of key-value (~dictionaries).e.g. criteria = {action_id = <John Doe’s id>, status = ‘pending’, program_id = <program A id>).

REST API endpoints examples

- POST /user/personal_background
  => this will add personal background information to the user profile.
- GET, PUT, DELETE /user/personal_background/{user_id}
  => this will get/update/delete personal background information of the user which id is specified.
- GET /users/find_by_physical_ability/{physical_ability}
  => this will get a list of all users which have the physical ability match the type specified in the 'physical_ability' passed as parameter.
- GET /users/find_by_criteria/{criteria}
  => this will get a list of users that met the specific search criteria set in 'criteria' parameter which is a map of key-value (~ dictionary). This way we don't need to write an extensive list of GETs to filter users by each of their attributes (such as GET /users/find_by_physical_ability).
- POST /company
  => this will create a company.
- GET, PUT, DELETE /company/{company_id}
  => this will get/update/delete the details of the company which id is specified.
- GET /companies
  => this will get the list of all companies.
- GET /companies/{company_id}
  => this will get the company which id is specified.
- GET /companies/find_by_country/{country}
  => this will get a list of companies in the specified country.
- GET /companies/find_by_criteria/{criteria}
  => this will get a list of companies that met the specific search criteria set in 'criteria' parameter which is a map of key-value (~ dictionary).
- POST /program
  => this will create a program.
- GET/PUT/DELETE /program/{program_id}
  => this will get/update/delete the details of the program with the specified id.
- GET /programs
  => this will get the list of all programs.
- GET /programs/{program_id}
  => this will get the program which id is specified.
- GET /programs/find_by_contact_type/{contact_type}
  => this will get a list of programs that have contact_type specified by the parameter.

- GET /programs/find_by_criteria/{criteria}
=> this will get a list of programs that met the specific search criteria set in 'criteria' parameter which is a map of key-value (~ dictionary).
- POST /mentorship_relation/send_request
=> this will create a mentoring relation request as well as send email notification to recipient about **MR** request
- GET, DELETE /mentorship_relation/{mentorship_relation_id}
=> this will get/delete the mentorship_relation with the specified id.
- GET /mentorship_relations
=> this will get the list of all **MR**s.
- PUT /mentorship_relations/{mentorship_relation_id}
/accept
=> this will allow the Company/Mentor/Mentee to accept an **MR** request.
- PUT /mentorship_relations/{mentorship_relation_id}
/reject
=> this will allow the Company/Mentor/Mentee to reject an **MR** request.
- PUT /mentorship_relations/{mentorship_relation_id}
/cancel
=> this will allow the Company/Mentor/Mentee to cancel an **MR** request.
- GET / mentorship_relations/current
=> this will get the list of current **MR** regardless of their 'state' for the logged in user.
- GET / mentorship_relations/find_by_criteria/{criteria}
=> this will get the list of **MR**s based on the 'criteria' passed on as the parameter.

Project Features/Enhancements

What are the features/enhancements that you plan for the summer with <your project>? (please use bullet points)

Note: From here onward any integration with Mentorship System will be referred to IMS

Application Features

Applicable to registered or non-registered users. No login required

- [**Required**] All Users can see a list of Mentoring Programs posted by Companies. A **Program** would contain the followings:
    - [**Required**] information on Program ID, Program Title, Start and End Dates, Program Location and Timezone, Program Description, minimum Requirements to be a Mentor and Mentee, Company/Mentor/mentee Responsibilities, Target Skill sets expected to be gained through the program, Payment Type (such as stipend for mentor and mentee, or none), Contact Type (boolean value for each of Face-to-face/Remote/Open to Global community), Resources Provided by the Company, Contact Person, Contact Number, Program Tags, Program Status (Open/In Progress/Closed). A Program with status In Progress or Closed will have a Mentor's and Mentee's name attached to them (that links to their MS User Profile) (IMS).
    - [**Required**] A **Company Portfolio** that would contain information on Company Name, About Us (brief description), Website link, Location and Timezone, Contact Number, Email, information on:
        - [**Future**] *(Data Analysis)* Company's demographic graph that shows the current proportion of employees based on the **Employee Background** *(see term reference).*
    - [**Required**] **Target Candidate** that allows the Company to specifically target people who are within the **Marginalised Groups** *(see term reference)* in the program they are offering. If the Program does not have any specific target, Target Candidate value will be None.
    - [**Required**] An IRC Channel space (like Zulip) where a Company, Mentor and Mentee who are in a relationship within the Program can communicate by posting questions, feedback, or anything related to the Program.
    - [**Nice to have**] **Ratings** - received from Company/Mentor/Mentee on overall user's experience throughout the Program (on the Program that is already Closed only).
    - [**Nice to have**] Company/Mentor/Mentee that are related to a program can give **Feedback** to each other on their performance (on the Program that is already Closed only).
    - [**Future**] All Users can post a comment inside the **Comment** box and view all comments.
- [**Future**] All Users can see blogs activity between users. A **Blog** would contain information on:

    - [**Required**] Blog Title, Blog Date, User Name, User Type (Company, Mentor, or Mentee), Blog Content, Blog tags, number of Likes, Comments.
- [**Nice to have**] All Users can see overview of lists of upcoming Community Events held by Companies. An **Even**t would contain information on:
    - [**Required**] Event ID, Event Name, Event Start and End Dates, Event Time and Timezone, Event Location, Event Description.
- [**Nice to have**] All Users can see **Highlights** in the form of the feedback received by a Program with Average Ratings 5/5 from its related Users.
- [**Future**] *(Data Analysis)* All Users can see a graph of the Tech Industry employees demographic which highlighted the proportion of the Marginalised Groups. This will be taken from the employment data of the Tech Industry in the country they live in.
- [**Future**] All Users can see a list of Top 10 Companies who at the time of viewing have the highest number of people in marginalised groups category listed as mentees in their mentoring programs.

Applicable only to Login User

- [**Required**] All Users can see the progress of the program which is shown by Tasks completion bar (IMS).
- [**Required**] All Users can see the Portfolio of any Company/Mentor/Mentee with the CRUD functionalities accessible only to the User whose the Portfolio belongs to (IMS).
- [**Required**] A Company can send Request to Mentor or Mentee to work on their Program (IMS).
- [**Required**] A Company can accept Request from Mentor or Mentee who wants to work on their Program(IMS).
- [**Required**] A Mentor/Mentee can send Request to Company to work on a Program (IMS).
- [**Required**] A Mentor/Mentee can accept Request from Company to work on a Program
- [**Required**] All Users can see the list of programs a Mentor or a Mentee has applied to/are currently working on/have completed (IMS).
- [**Required**] A Mentor/Mentee must give details on their **User Background** in their Profile (IMS). This User Background would have:
    - [**Required**] range of selections on Education level, Work Experience (time), Gender, Age, Sexual Orientation, Race, Physical (dis)Ability, Income level, Religion. (An option to select 'Prefer not to say' is in one of the selections).
    - [**Required**] an option to be disclosed publicly (Unhide) or kept as private (Hide).

- [**Required**] A Mentor/Mentee can access the Mentorship System UI without having to login again (IMS).
- [**Nice to have**] A Company can give Ratings on the Mentor and the Mentee who have completed their program (IMS).
- [**Nice to have**] A Mentee who has completed a program can give Ratings to the Program and the Mentor (IMS).
- [**Nice to have**] A Mentor who has completed a program can give Ratings to the Program and the Mentee (IMS).

Application Feature Constraints

- [**Required**] One Program can only be in one relationship with a Mentor and a Mentee.
    - [**Future**] 1> A Program can have more than one Mentor but can only have one Mentee (IMS).
    - [**Future**] 2> A Program can have more than one Mentor and more than one Mentee (IMS).

Potential Timeline for the Features/Enhancements

(Please note that this timeline is only meant to be a guide for your work throughout the program, your goals and expectations will be adjusted during the Community Bonding Period after discussion with your mentors and admins.)

Period (Community bonding, First Coding Phase, etc.)	Milestone	Subtasks	Due Date
Community Bonding (04/05 - 31/05)	Discussion with mentors on technical requirements and app features	- Tech Stack - Minimum Viable Product (MVP) - App flow - App architecture	04/05 -07/05 (4 days)
	Setup Initial documentation	- Project Scope Requirements - Version control + Wiki + Reports template	08/05 - 10/05 (3days)
	Review Wireframe +	- Discuss structure,	11/05 - 13/05

	App flow	(reusable) components, assets, colors, interactions, path	(3 days)
	Create GUI Prototype	- Draft Prototype - Discuss w mentors - Adjust from feedback	14/05 - 17/05 (4 days)
	Backend Design and setup (environment, dependencies, models, container) (IMS)	- ERD (Entity Relationship Diagram) - UML Class Diagram - Review REST API endpoints+CRUD functions - Review Backend Design - Setup\update Backend Environment (IMS) - Setup/update local CI and Code Quality tools	18/05 - 30/05 (13 days)
	Time off		31/05 - 31/-5 (1 day)
First Coding Phase (01/06 - 28/06)	Week 1. Start Backend dev	- Create and populate Data Models Classes - Update local/remote? servers (REST APIs + endpoints) (IMS) - Update Wiki - Weekly Report	01/06 - 08/06 (8 days)
	Week 2a. Continue backend dev preparation from week 1 above.	- Create test cases	09/06 - 11/06 (3 days)
	Week 2b. User stories: As a user, I would like to see a public landing page (Homepage) so that I can use it as a starting point to get to BridgeInTech	- Homepage backend - Run test cases/fix bugs - Homepage UI (fig.1.3) - Consumption Homepage endpoints - Update Wiki - Weekly Report	12/06 - 15/06 (4 days)
	Week 3a. User stories: As a user, I would like to register/login so	- Register/Login backend - Run test cases/fix bugs - Register/Login UI (fig.1.1	16/06 - 19/06 (3.5 days)

	that I could see my dashboard.	& 1.2) - Consumption Register/Login endpoints	
	Week 3b. User stories: As a user, I would like to see the list of all companies involved in BridgeInTech, so that I could go through the list and select which company I should work with in the mentoring program.	- Companies list backend - Run test cases/fix bugs - Companies list UI (fig.1.4) - Consumption Companies list endpoints - Update Wiki - Weekly Report	19/06 - 22/06 (3.5 days)
	Week 4a. User stories: As a user, I would like to see the list of all programs offered by BridgeInTech, so that I could go through the list and select which program I should apply to.	- Programs list backend - Run test cases/fix bugs - Programs list UI (fig.2.1) - Consumption Programs list endpoints	23/06 - 25/06 (3 days)
	Week 4b. User stories: As a user, I would like to see the list of all mentors and mentees, so that I could see who I should send a request to work with in a mentoring program.	- Mentors/Mentees list backend - Run test cases/fix bugs - Mentors/Mentees list UI (fig.2.2) - Consumption Mentors/Mentees endpoints - Update Wiki - Weekly Report	26/06 - 28/06 (3 days)

Phase 1 Evaluation (29/06 - 03/07) *Buffer time

Second Coding (04/07 - 26/07)	Week 1a. User stories: As a user, I would like to see the Company Portfolio page, so that I could get	- Company Portfolio backend - Run test cases/fix bugs - Company Portfolio UI (fig.2.3) - Consumption Company	04/07 - 06/07 (3 days)

	information about the company.	Portfolio endpoints	
	Week 1b. User stories: As a user, I would like to see the Program Details page, so that I could get informed about what the program is about and apply to it if I'm interested.	- Program Details backend - Run test cases/fix bugs - Program Details UI (fig.3.1) - Consumption Program Details endpoints - Update Wiki - Weekly Report	07/07 - 09/07 (3 days)
	Week 2a. User stories: As a user, I would like to see the Mentor/Mentee Portfolio page so that I can get information about the Mentor/Mentee.	- Mentor/Mentee Portfolio backend - Run test cases/fix bugs - Mentor/Mentee Portfolio UI (fig.3.2) - Consumption Mentor//Mentee Portfolio endpoints	10/07 - 12/07 (3 days)
	Week 2b. User stories: As a company user, I would like to be able to create a Company Profile, so that I can showcase who we are as a company.	- Create Company Profile backend - Create Company Profile UI (fig.3.3) - Run test cases/fix bugs - Consumption Create Company Profile endpoints - Update Wiki - Weekly Report	13/07 - 15/-7 (3 days)
	Week 3a. User stories: As a company user, I would like to be able to Create a Program, so that I could offer the program to mentors and mentees	- Create Program backend - Run test cases/fix bugs - Create Program UI (fig.4.1-4.3) - Consumption Create Program endpoints	16/07 - 18/07 (3 days)
	Week 3b. User stories:	- Company Request backend	19/07 - 21/07 (3 days)

	As a company user, I would like to be able to send a request to a mentor or mentee, so that I could ask them to work in the program I'm offering.	- Run test cases/fix bugs - Company Request Form UI (fig 4.4) - Consumption Company Request endpoints - Update Wiki - Weekly Report	
	Week 4a. User stories: As a mentor/mentee user, I would like to send a request to a company on one of its programs, so that I could work on that mentoring program I'm interested in.	- Application form backend - Run test cases/fix bugs - Application Form UI (fig.4.5) - Consumption Application Form endpoints	22/07 - 24/07 (3 days)
	Week 4b. User stories: As a mentor/mentee user, I would like to create a profile, so that I could give the BridgeInTech community some information about me.	- Create Profile backend - Run test cases/fix bugs - Create Profile UI (fig.4.6) - Consumption Create Profile endpoints - Update Wiki - Weekly Report	25/07 - 27/07 (3 days)

Phase 2 Evaluation (27/07 - 31/07) *Buffer time

Third Coding Phase (01/08 - 31/08) Sem 2 Uni Start date: (Tentative) 03/08 - enrolled in one unit, so it shouldn't affect my work in GSoC)	Week 1. Unit integration (wk 0 uni)	- Run unit tests and fix bugs - Update Wiki - Weekly Report	01/08 - 09/08
	Week 2. System integration (wk 1 uni)	- Run system tests and fix bugs - Update Wiki - Weekly Report	10/08 - 16/08
	Week 3. Project Demo. (wk 2 uni)	- Review App with mentors - Make necessary adjustment - Update Wiki - Weekly Report	17/08 - 23/08

	Final Week + Mentor Evaluation (wk 3 uni)	- Wrap up, review and adjust Wiki and Weekly Reports - Final Report	24/08 - 31/08

Final Evaluation (31/08 - 07/09)

What is your backup plan for time management in the event of unseen difficulties that you might encounter? (Eg: computer breaking down, family emergency etc.)
I would immediately inform my mentors/admin to negotiate how I could best compensate for the time loss so to pick up the pace as quickly as I can to ensure I will not fall behind the targeted timeline. In brief, I am committed to do the best I can.

Optional Question

I'm a passionate learner who wants to give back to the community while learning.

References

- Back4App (2019). *GraphQL vs REST*. [online] Medium. Available at: https://medium.com/@back4apps/graphql-vs-rest-62a3d6c2021d [Accessed 22 Mar. 2020].
- Baldwin, M.L. and Marcus, S.C. (2007). Labor Market Outcomes of Persons with Mental Disorders. *Industrial Relations*, 46(3), pp.481–510. Available at: https://doi-org.ezproxy.lib.monash.edu.au/10.1111/j.1468-232X.2007.00478.x [Accessed 21 Mar.2020]
- Clay, R.A. (2012). 20 years of advancing mental and behavioral health. *https://www.apa.org*. [online] Dec. Available at: http://www.apa.org/monitor/2012/12/20-years [Accessed 21 Mar. 2020].
- Coussement, J. (2016). *Replacing EAV with JSONB in PostgreSQL*. [online] coussej.github.io. Available at: https://coussej.github.io/2016/01/14/Replacing-EAV-with-JSONB-in-PostgreSQL/ [Accessed 28 Mar. 2020].
- FREEMANAGERMENTORS.COM (2019a). *Free Mentorship For Managers & Leaders* -. [online] www.freemanagermentors.com. Available at: https://www.freemanagermentors.com/ [Accessed 18 Mar. 2020].
- Girish, R. (2017). *Progressive Web App VS Responsive Web App*. [online] Techathlon. Available at: https://techathlon.com/progressive-web-app-vs-responsive-web-app/ [Accessed 18 Mar. 2020].
- Guru99 (2019b). *SQL vs NoSQL: What's the difference?* [online] Guru99.com. Available at: https://www.guru99.com/sql-vs-nosql.html [Accessed 22 Mar. 2020].
- Halliday, L. (2015). Unleash the power of storing JSON in Postgres - via @codeship. [online] Rollout Blog. Available at: https://rollout.io/blog/unleash-the-power-of-storing-json-in-postgres/ [Accessed 28 Mar. 2020].

- ➢ Haustant, A. (2014a). *Request Parsing — Flask-RESTPlus 0.13.0 documentation*. [online] flask-restplus.readthedocs.io. Available at: https://flask-restplus.readthedocs.io/en/stable/parsing.html [Accessed 22 Mar. 2020].

- ➢ Haustant, A. (2014b). *Swagger documentation — Flask-RESTPlus 0.13.0 documentation*. [online] flask-restplus.readthedocs.io. Available at: https://flask-restplus.readthedocs.io/en/stable/swagger.html [Accessed 22 Mar. 2020].

- ➢ KALEIDOSCOPE (2020a). *Home*. [online] Kaleidoscope. Available at: https://www.kaleidoscopeinitiative.com.au/ [Accessed 18 Mar. 2020].

- ➢ Machanda, A. (2019). *Insights - Web and Mobile Development Services and Solutions*. [online] Insights - Web and Mobile Development Services and Solutions. Available at: https://www.netsolutions.com/insights/cross-platform-app-frameworks-in-2019/ [Accessed 18 Mar. 2020].
- ➢ Markou, E. (2018). *Storing JSON in PostgreSQL: Step by Step Guide | Blendo.co*. [online] Blendo. Available at: https://www.blendo.co/blog/storing-json-in-postgresql/ [Accessed 28 Mar. 2020].
- ➢ Matteson, S. (2019). *How to close the tech skills gap*. [online] TechRepublic. Available at: https://www.techrepublic.com/article/how-to-close-the-tech-skills-gap/ [Accessed 12 Mar. 2020].
- ➢ MERCY CARE (2020b). *Aged Care Perth - Childcare Perth - NDIS - MercyCare*. [online] Mercy Care. Available at: https://www.mercycare.com.au/ [Accessed 18 Mar. 2020].

- ➢ OECD (2015). *Fit mind, fit job : evidence to practice in mental health and work.* [online] Paris, France: OECD. Available at: https://doi-org.ezproxy.lib.monash.edu.au/10.1787/9789264228283-en [Accessed 22 Mar. 2020].

➢ O'Hara, M. (2007). *Mental health and employment: the facts*. [online] the Guardian. Available at: https://www.theguardian.com/society/2007/nov/14/mentalhealth2 [Accessed 21 Mar. 2020].

➢ Ozar, B. (2011). *How to Design Multi-Client Databases*. [online] Brent Ozar Unlimited®. Available at: https://www.brentozar.com/archive/2011/06/how-design-multiclient-databases/ [Accessed 20 Mar. 2020].

➢ Pretty Printed (2016a). *Using Database Schemas in Flask-SQLAlchemy*. *YouTube*. Available at: https://www.youtube.com/watch?v=V3IX_M3fwd0 [Accessed 21 Mar. 2020].

➢ Python, R. (2020). *Working With JSON Data in Python – Real Python*. [online] realpython.com. Available at: https://realpython.com/python-json/ [Accessed 28 Mar. 2020].

➢ Resnisky, H. (2018). *Machine Learning in PostgreSQL Part 1: Kmeans clustering*. [online] Cybertec. Available at: https://www.cybertec-postgresql.com/en/machine-learning-in-postgresql-part-1-kmeans-clustering/ [Accessed 20 Mar. 2020].

➢ Ringer, C. (2015). *PostgreSQL anti-patterns: Unnecessary json*. [online] 2ndQuadrant | PostgreSQL. Available at: https://www.2ndquadrant.com/en/blog/postgresql-anti-patterns-unnecessary-jsonhstore-dynamic-columns/ [Accessed 28 Mar. 2020].

➢ Robinson, D. (2016). Heap: When To Avoid JSONB In A PostgreSQL Schema. [online] Heap. Available at: https://heap.io/blog/engineering/when-to-avoid-jsonb-in-a-postgresql-schema [Accessed 28 Mar. 2020].

➢ SCORE.ORG (2016b). *Find a Mentor*. [online] www.score.org. Available at: http://www.score.org/find-mentor [Accessed 18 Mar. 2020].

➢ TestDriven.io (2019c). *Authentication with Flask, React, and Docker | TestDriven.io*. [online] testdriven.io. Available at: https://testdriven.io/courses/auth-flask-react/ [Accessed Dec. 2019].

➢ TestDriven.io (2019d). *Developing a RESTful API with Python, Flask, Docker, and Pytest | TestDriven.io*. [online] testdriven.io. Available at: https://testdriven.io/courses/tdd-flask/ [Accessed Dec. 2019].

➢ https://stackoverflow.com/questions/386294/what-is-the-maximum-length-of-a-valid-email-address

➢ https://stackoverflow.com/questions/1423195/what-is-the-actual-minimum-length-of-an-email-address-as-defined-by-the-ietf

➢ https://stackoverflow.com/questions/723587/whats-the-longest-possible-worldwide-phone-number-i-should-consider-in-sql-varc

➢ https://stackoverflow.com/questions/14894899/what-is-the-minimum-length-of-a-valid-international-phone-number

➢ https://stackoverflow.com/questions/417142/what-is-the-maximum-length-of-a-url-in-different-browsers

➢ https://medium.com/@cancerian0684/which-data-type-would-you-choose-for-storing-currency-values-like-trading-price-dd7489e7a439

➢ https://www.postgresqltutorial.com/postgresql-to_number/

➢ https://medium.com/@cancerian0684/which-data-type-would-you-choose-for-storing-currency-values-like-trading-price-dd7489e7a439

➢ https://github.com/knex/knex/issues/1869 => SQLite multiple schemas alternative

➢ https://stackoverflow.com/questions/33960762/does-sqlite-support-multiple-schematas-within-the-same-database => SQLite does not allow multiple schemas

➢ https://stackoverflow.com/questions/28951786/postgresql-multiple-database-vs-multiple-schemas => Postgres allow multiple schema

Term reference

➢ **Marginalised Groups**: those who are considered to be among the minority in regard to the Employee Background category mentioned above.

➢ **Employee Background** categories: Gender, Age, Sexual Orientation, Race, Physical (dis)Ability, Income level, Religion, Education level, Work Experience (length of time).

Note: Each of these groups will have range of selections (e.g. Gender: Male, Female, Other; Sexual Orientation: Heterosexual, LGBT++; etc...). All of these categories will have an option of ‘Prefer not to say’ in them.