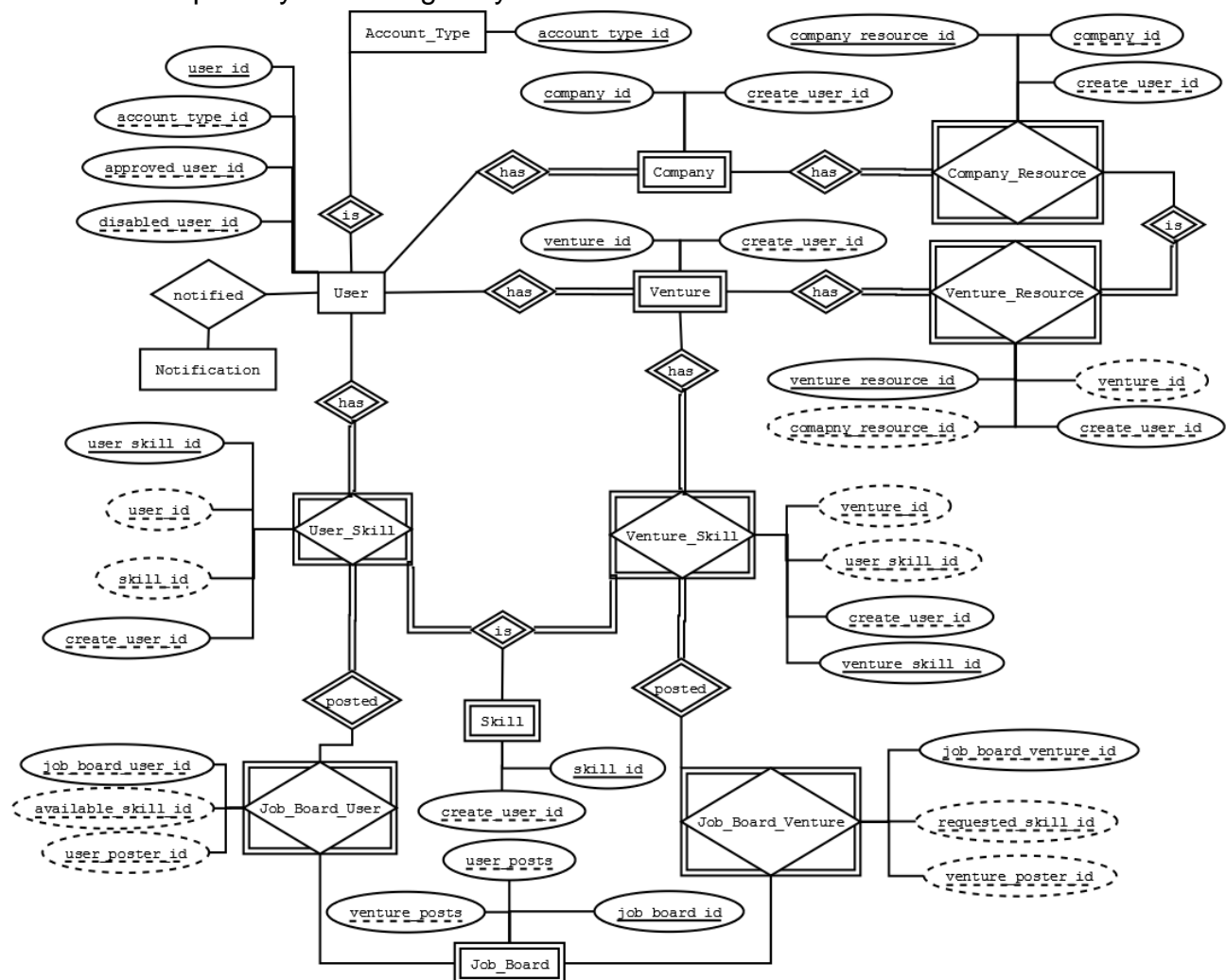


3.1. Functional Requirements (Bliss)

The requirements necessary to build a functional prototype that will prove the viability of a solution to Strome Entrepreneurial Center's lack of ability to make, maintain, and track connections. Each of the following sections will address specific portions of the digital solution, how they will be used, and their constraints.

1. Database (Bliss)

Implementing a PostgreSQL database provides a structured format for saving, editing, and accessing data for a given concept of the prototype. Data elements stored within the database will be constrained to specific data types and formats. The following schema shows the tables and their primary and foreign keys.



1. User Table (Strong)

Data on user shall be saved in the following format:

Attribute	Datatype	Key	Requirement	Null	Notes
user_id	int	Primary key	Must auto-increment starting at 1	Not Null	
last_name	varchar			Not Null	
first_name	varchar			Not Null	
email	varchar		Must be unique	Not Null	
password	varchar			Not Null	
register_dttm	date			Null	Populated after email verification
create_dttm	date		Must default to SYSDATE when record is created	Not Null	
account_type_id	int	Foreign key to Account_Type (account_type_id)		Not Null	
approved_dttm	date			Null	Null value assumes user is not approved
approved_user_id	int	Foreign key to User (user_id)		Null	
disabled_dttm	date			Null	Null value assumes user is active
disabled_user_id	int	Foreign key to User (user_id)		Null	
gcm_token	varchar				

Space left intentionally blank.

2. User Account Types (Strong)

Data on user_account shall be saved in the following format:

Attribute	Datatype	Key	Requirement	Null	Notes
account_type_id	int	Primary key	Must auto-increment starting at 1	Not Null	
name	varchar		Must be unique	Not Null	
description	varchar			Null	

3. Ventures (Bliss)

Data on venture shall be saved in the following format:

Attribute	Datatype	Key	Requirement	Null	Notes
venture_id	int	Primary key	Must auto-increment starting at 1	Not Null	
description	varchar			Null	
create_user_id	int	Foreign key to User (user_id)		Not Null	
create_dttm	date		Must default to SYSDATE when record is created	Not Null	
public_info	varchar			Null	
approved_dttm	date			Null	
approved_user_id	int	Foreign key to User (user_id)		Null	

Space left intentionally blank.

4. Skills (Bliss)

Data on skill shall be saved in the following format:

Attribute	Datatype	Key	Requirement	Null	Notes
skill_id	int	Primary key	Must auto-increment starting at 1	Not Null	
name	varchar		Must be unique	Not Null	i.e. Web Developer
description	varchar			Null	
create_dttm	date		Must default to SYSDATE when record is created	Not Null	
create_user_id	int	Foreign key to User (user_id)		Not Null	

5. User Skill (Bliss)

Data on user_skill shall be saved in the following format:

Attribute	Datatype	Key	Requirement	Null	Notes
user_skill_id	int	Primary key	Must auto-increment starting at 1	Not Null	
user_id	int	Foreign key to User (user_id), partial key (user_id, skill_id)		Not Null	
skill_id	int	Foreign key to Skill (skill_id), partial key (user_id, skill_id)		Not Null	
create_dttm	date		Must default to SYSDATE when record is created	Not Null	
create_user_id	int	Foreign key to User (user_id)		Not Null	

Space left intentionally blank.

6. Venture Skill (Bliss)

Data on venture_skill shall be saved in the following format:

Attribute	Datatype	Key	Requirements	Null	Notes
venture_skill_id	int	Primary key	Must auto-increment starting at 1	Not Null	
venture_id	int	Foreign key to Venture (venture_id), partial key (venture_id, user_skill_id)		Not Null	
user_skill_id	int	Foreign key to User_Skill (user_skill_id), partial key (venture_id, user_skill_id)		Not Null	
create_dttm	date		Must default to SYSDATE when record is created	Not Null	
create_user_id	int	Foreign key to User (user_id)		Not Null	

7. Companies (Bliss)

Data on company shall be saved in the following format:

Attribute	Datatype	Key	Requirements	Null	Notes
company_id	int	Primary key	Must auto-increment starting at 1	Not Null	
name	varchar		Must be unique	Not Null	i.e. Olive Garden
description	varchar			Null	
create_dttm	date		Must default to SYSDATE when record is created	Not Null	
create_user_id	int	Foreign key to User (user_id)		Not Null	

Space left intentionally blank.

8. Company Resources (Bliss)

Data on company_resource shall be saved in the following format:

Attribute	Datatype	Key	Requirements	Null	Notes
company_resource_id	int	Primary key	Must auto-increment starting at 1	Not Null	
name	varchar		Must be unique	Not Null	i.e. Kitchen Space
description	varchar				
company_id	int	Foreign key to Company (company_id)			
create_dttm	date		Must default to SYSDATE when record is created		
create_user_id	int	Foreign key to User (user_id)			

9. Venture Resources (Bliss)

Data on venture_resource shall be saved in the following format:

Attribute	Datatype	Key	Requirements	Null	Notes
venture_resource_id	int	Primary key	Must auto-increment starting at 1	Not Null	
venture_id	int	Foreign key to Venture (venture_id), partial key (venture_id, resource_id)		Not Null	
company_resource_id	int	Foreign key to Company_Resource (company_resource_id), partial key (venture_id, company_resource_id)		Not Null	
create_dttm	date		Must default to SYSDATE when record is created	Not Null	
create_user_id	int	Foreign key to User (user_id)		Not Null	

Space left intentionally blank.

10. Job Board (Strong)

Data on job_board shall be saved in the following format:

Attribute	Datatype	Key	Requirements	Null	Notes
job_board_id	int	Primary key	Most auto-increment starting at 1	Not Null	
user_posts	int	Foreign key to Job_Board_Users (job_board_user_id)			
venture_posts	int	Foreign key to Job_Board_Venture (job_board_venture_id)			

11. Job Board Users (Strong)

Data on notification shall be saved in the following format:

Attribute	Datatype	Key	Requirements	Null	Notes
job_board_user_id	int	Foreign key to Job_Board(job_board_id)	Most auto-increment starting at 1	Not Null	
available_skill_id	int	Foreign key to Skill(skill_id), partial key (user_id, skill_id)			
user_poster_id	int	Foreign key to User(user_id), partial key (skill_id, user_id)			

12. Job Board Ventures (Strong)

Data on notification shall be saved in the following format:

Attribute	Datatype	Key	Requirements	Null	Notes
job_board_venture_id	int	Foreign key to Job_Board(job_board_id)	Most auto-increment starting at 1	Not Null	
requested_skill_id	int	Foreign key to Skill(skill_id), partial key (venture_id, skill_id)			
venture_poster_id	int	Foreign key to User(user_id), partial key (skill_id, venture_id)			

Space left intentionally blank.

13. Notification (Strong)

Data on notification shall be saved in the following format:

Attribute	Datatype	Key	Requirements	Null	Notes
notification_id	int		Must auto-increment starting at 1	Not Null	
message	varchar			Not Null	
read_dttm	date			Null	Null values assumes message not read
create_dttm	date		Must default to SYSDATE when record is created	Not Null	
create_user_id	int	Foreign key to User (user_id)		Not Null	
to_user_id	int	Foreign key to User (user_id)		Not Null	

2. Web Service (API) (John S)

The Web service shall support the front end of the application by providing an interface for common functions of the database. The web service shall restrict access to data contained in the database based on the specifications of the Access Control Table.

1. User Services

1. User Registration/Create
2. User Login
3. Profile Lookup
4. User Edit
5. Register mobile application
6. User Profile Associate Skill
7. User Profile Remove Skill
8. User Profile Notifications Lookup
9. Retrieve Users with Skill
10. Approve User

2. Venture Services

1. Create New Venture
2. Lookup Venture
3. Edit Venture Information
4. Request Meeting
5. Add User Membership
6. Remove User Membership

3. Companies

1. Create Company

2. Modify Company
3. Offer Resource
4. Remove Resource Offering
5. Add Resource Request
6. Remove Resource Request
4. Resources
 1. New Resource
 2. New Resource Request
 3. Search for Resource
5. Notifications (John S)
 1. Create New Notification
Shall create a new unread notification targeting a list of users.
 2. Get Unread Notifications
Shall provide the authenticated user with a list of notifications that have not yet been read by the user.
 3. Get All Notifications
Shall retrieve all notifications for the authenticated user with a flag for whether or not that user has read the notification.
 4. Read Notification
Shall mark a pending notification of the authenticated user as read.
6. Admin
 1. Add Domain Whitelist
 2. Remove Domain Whitelist
 3. Add Test User
 4. Deactivate Test User

	SEC Staff	ODU Student	External Company
General			
Automatic Registration			
Send Notifications			
View Notifications			
Create User			
Add SEC Staff			
Add/Remove Domain Whitelist			
Student			
View Student Profile			

View Student Skills			
Add Skill			
Delete Skill			
Venture			
Search Venture			
View Venture			
Request Venture			
View Venture Skills			
Add/Remove Members			
Edit Venture Details			
Approve Venture			
Delete Venture			
Approve Join Venture Request			
Request Skills/Resoures			
Request to Join Venture			
Company			
Search Companies			
View Company Profile			
Edit Company Profile			
Add Company Resource			
Remove Company Resource			
Approve Company			
Delete Company Profile			
Job Board			
Search Job Board			
View Job Board Listing			
Post to Job Board			
Remove Posted Listing			

Report Spam			
View Spam Reports			
Remove Job Board Listing			

Figure ??: Access Control Table, green indicates the action is accepted, red indicates it is not allowed, and yellow indicates a scenario with pre-requisites.

3. Web Interface (John S)

The web interface shall provide users with a front end interface for interacting with data accessible via the web service (Section 3.1.2). The interface must be well optimized and tested to work with mobile devices. The interface must also adapt in order to only show features allowed by the Access Control Table.

1. Users

1. Registration
2. Login
3. Profile
4. Profile Edit

2. Ventures

1. Create
2. Modify
3. Profile

3. Job Board

1. Resource Search
2. Resource Request
3. Resource Offer

4. Companies

1. Create
2. Modify
3. Profile

5. Notifications (John S)

1. Unread Notifications
Shall be indicated by an icon and number on all pages that can be expanded in order to mark notifications as read.
2. Read notifications
A notifications page shall allow the user to view all notifications including previously read notifications. Read/Unread notifications must be indicated as such.

6. Administration

1. Edit User Whitelist
2. Test User Create
3. Send Notification

4. Notifications (John S)

Notifications shall alert users that something needs their attention after they have

closed the application. There shall be two methods outside of the user interface mentioned in section 3.1.3.5 that will service this purpose.

1. Digest Email

1. Automatically Subscribed

Notifications by email shall be sent as a digest email which includes all unread notifications once per week.

2. Unsubscribe

All emails shall include a direct link to unsubscribe from the weekly digest emails.

2. Push

Users who access the service using the downloadable android application shall receive notifications via push notification service.

1. Register Mobile Device

On first launch the device shall be registered to receive pushes from the web service in real time as events occur.

2. Receive Notification

Notifications sent by the web services shall be received by the device to display a native notification which on click links to the relevant page in the application and marks the notification as read.