3 RocStar Specific Requirements

This section details the functional requirements for RocStar. It will address the frontend and backend requirements, to include the algorithms and database specifications.

3.1 Functional Requirements

3.1.1 Frontend Functional Requirements

3.1.1.1 Application User Requirements (Jason Howse)

RocStar shall have 5 user account accounts. These accounts shall be Hospital staff account, Family account, Volunteer account, Team Leader account, RSF staff account, and tester account.

3.1.1.1.1 The RocStar shall provide a specific user account for Hospital/Child Life Specialists. Users with this account shall have access to the following.

a. Hospital staff accounts shall provide availability of requests to RSF by hospital staff for more ready bags.
b. Hospital staff accounts shall provide users with referral forms to refer families to RSF.
c. Hospital staff accounts shall provide status update for referrals.

3.1.1.1.2 The RocStar shall provide a specific user account for the families of children suffering from cancer. Users with this account shall have access to the following.

a. Family accounts shall allow users to provide registration of ready bags.
b. Family accounts shall provide a donate button.
c. Family accounts shall have a specific “kid zone” from family features.
d. Family accounts shall be able to view a status report for build projects.
e. Family accounts shall provide waivers.
f. Family accounts shall provide inspirational notifications.
j. Family accounts may have preloaded movies and games.

3.1.1.3 The RocStar shall provide a specific user account for volunteers. Users with this account shall have access to the following.
   a. Volunteer accounts shall have access to a calendar of events.
   b. Volunteer accounts shall push notification reminders.
   c. Volunteer accounts shall provide access to the online store.
   d. Volunteer accounts shall provide a donate button.
   e. Volunteer accounts should provide a checklist for eligible events.
   f. Volunteer accounts should have access to a volunteer manual.
   g. Volunteer accounts may provide a volunteer waiver.

3.1.1.4 The RocStar shall provide a specific user account for team leaders. Users with this account shall have access to the following.
   a. Team leader accounts shall provide the same access as Volunteer accounts 3.1.1.3.
   b. Team leader accounts shall provide instant message support to project volunteers.
   c. Team leader accounts shall provide reminders to volunteer and Team leader accounts.
   d. Team leader accounts shall provide project checklists for activities and supplies.
   e. Team leader accounts may provide a drop pin in a map option when each project is finished.

3.1.1.5 The RocStar shall provide a specific user account for RSF Staff. Users with this account shall have access to the following.
   a. RSF Staff accounts shall allow users to receive applications/requests to and from program coordinators.
   b. RSF Staff accounts shall allow users access to calendar events.
   c. RSF Staff accounts shall allow users to access the NeonCRM database.
   d. RSF Staff accounts shall allow users to coordinate ready bag logistics.
   e. RSF Staff accounts shall allow users to coordinate with all other users of RocStar.
   f. RSF Staff accounts shall allow users to determine which hospitals have access to ready bags.
g. RSF Staff accounts should allow users access to the file system.
h. RSF Staff accounts should allow users to track Ready Bag inventory.
i. RSF Staff accounts should allow users to provide an overview of recipients.
j. RSF Staff accounts should allow users to overview of projects.

3.1.1.6 The RocStar shall provide a specific user account for testing.
Users with this account shall have access to the following.
a. Tester accounts shall...

3.1.1.2 Create/Edit User Profiles (Cameron Covington)
RocStar shall allow Roc Solid Foundation staff, volunteers, family members, and a designated tester to create and edit his user profile. The system shall provide the ability to input user information, as well as edit that information later via the GUI. After entering his information, the user shall be prompted to save his user profile information in the RocStar application. The GUI provides the user the ability to edit his profile information. Before the user begins editing his information, RocStar shall populate the user's information from the MySQL database in the relevant fields. After editing the necessary fields, the user is prompted to save his updated profile information. After updating the user's information in the MySQL database, RocStar shall also push those changes to Roc Solid Foundation's customer relationship management system, NeonCRM. The following functional requirements shall be met:

1. RocStar shall integrate with the MySQL database to store the user's information. (See requirement 3.1.2.3.1.1). User information shall include:

   A. Name
   
   B. Address
      - Street
      - City
      - State
      - Zip
      - Apartment Number (if applicable)
   
   C. Phone number
   
   D. Name of child with cancer (if applicable)
3.1.1.2.1 RocStar shall allow the user the ability to edit and update his user profile. The steps necessary for editing and saving the user profile information shall include:

A. User chooses to edit his user profile
B. RocStar automatically displays his saved user profile information from the database
C. User edits the necessary fields and clicks save
D. RocStar shall update his user profile information in the database to reflect his changes.

3.1.1.2.2 RocStar shall allow the user the ability to delete his user profile. The steps the user shall take to remove his user profile shall include:

A. User chooses to edit his user information
B. At the bottom of the edit screen, RocStar shall display a UI element to give the user the ability to delete his user profile.
C. When the user chooses to delete his user profile, the RocStar application will purge his user profile information from the database.

3.1.1.3 User Authentication and Authorization (Roland Mulligan)

The RocStar System shall provide management of system access, including user account creation, and user login and password management.

3.1.1.3.1 The System shall allow public users to access public parts of the System. The following functional requirements shall be met:

1. Provide public users with access to the online store
2. Provide public users with access to the donations page
3.1.1.3.2 The System shall allow users to gain access to authorized sections by a successful login. Logins shall use the Password Authentication Protocol. The following functional requirements shall be met:

3.1.1.3.2.1 Each user shall have a unique username

3.1.1.3.2.2 Each user shall have a unique password

3.1.1.3.2.2.1 Each password shall be at least eight characters long

3.1.1.3.2.2.2 Each password shall contain at least one uppercase letter

3.1.1.3.2.2.3 Each password shall contain at least one lowercase letter

3.1.1.3.2.2.4 Each password shall contain at least one numerical value

3.1.1.3.2.3 The login page must keep the password from being viewable in the password field

3.1.1.3.3 The System shall provide a way for users to retrieve forgotten passwords

3.1.1.3.4 The System shall allow users to change a password. When changing a password, the following functional requirements shall be met:

3.1.1.3.4.1 The System shall require a separate page for changing a password, without the ability to change other settings on the same page

3.1.1.3.4.2 The System shall require an entry with correct validation of current password in order to change password

3.1.1.3.4.2.1 Two fields shall be used to ensure user typed new password correctly

3.1.1.3.5 The System shall limit user access to specific parts of the system dependent on user type; e.g., volunteers can only access volunteer related pages
3.1.1.3.6 The System shall restrict certain functions to authorized personnel only; e.g., volunteers would have read-only permission for project pages, whereas team leaders would also have write permission

**ACCESS CONTROL TABLE**

<table>
<thead>
<tr>
<th></th>
<th>Admin/Tester</th>
<th>RSF Staff</th>
<th>Team Leader</th>
<th>Volunteer</th>
<th>Family/Child</th>
<th>Hospital Staff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Store</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Donations Page</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Events Calendar</td>
<td>View, Edit, Delete</td>
<td>View</td>
<td>View</td>
<td>View</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Projects Page</td>
<td>View, Edit, Delete</td>
<td>View</td>
<td>View</td>
<td>View</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Project Status</td>
<td>View, Edit, Delete</td>
<td>View</td>
<td>View</td>
<td>View</td>
<td>View</td>
<td>No</td>
</tr>
<tr>
<td>Electronic Referrals</td>
<td>View, Edit, Delete</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>View, Edit, Delete</td>
</tr>
<tr>
<td>Referral Status</td>
<td>View, Edit, Delete</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>View</td>
</tr>
<tr>
<td>Order Ready Bag</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RSF Waiver</td>
<td>View, Edit, Delete</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>View, Edit, Delete</td>
<td>No</td>
</tr>
<tr>
<td>Volunteer Waiver</td>
<td>View, Edit, Delete</td>
<td>No</td>
<td>No</td>
<td>View, Edit, Delete</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Volunteer Manual</td>
<td>View, Edit, Delete</td>
<td>No</td>
<td>No</td>
<td>View</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Family Page</td>
<td>View, Edit, Delete</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>View, Edit, Delete</td>
<td>No</td>
</tr>
<tr>
<td>Select Build</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Register Ready Bag</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
3.1.1.4 Electronically Fillable Hospital to Roc Solid Foundation Forms (James Rice)

RocStar must provide servicing hospitals an electronically fillable form to introduce patients to the Roc Solid Foundation.

3.1.1.4.1 The form(s) must allow an authorized user to enter and submit the necessary information for storage in the database (see requirement 3.1.2.3.6, 3.1.2.3.10, and 3.1.2.3.13). RocStar shall provide the following functional requirements:

3.1.1.4.1.1 the capability to input data.
3.1.1.4.1.2 the capability to edit data.
3.1.1.4.1.3 the capability to remove data.
3.1.1.4.1.4 the capability to save data.

3.1.1.4.2 The form must be comprised of the data entry elements of the child's, family, and physician’s information, as indicated.

RocStar must provide the following data entry functionality:

3.1.1.4.2.1 Child's Information
a. Name
b. Date of birth
c. Sex
d. Type of cancer
e. Attending hospital
f. Hospital address
g. Additional medical considerations (e.g., wheelchair, oxygen, etc.)

3.1.1.4.2.2 Family Information
a. Parent/legal guardian’s name (x2)
b. Address
c. Primary/secondary phone numbers
d. E-mail address

3.1.1.4.2.3 Physician’s Information
a. Name
b. Hospital’s name
c. Address
3.1.5 Mobile Application Requirements (Jason Howse)
RocStar shall have an Android application apk and Apple application ipa so users can conveniently access RocStar from mobile devices. The RocStar Mobile Application (RSMA) shall accessible on all prescribed devices.

3.1.5.1 The RSMA shall be available on mobile tablets and mobile phones.
   a. The RSMA shall be available on all Android devices at a minimum of Android version 4.1 API 16 (Jelly Bean).
   b. The RSMA shall be available on 95.2% of Android devices.
   c. The RSMA shall be available on iOS Devices.

3.1.2 Backend Functional Requirements

3.1.2.1 RocStar Admin Console (Joshua Harris)
RocStar shall provide an interface for Roc Solid Foundation staff and admin in order to meet these functional requirements:

- Receive and process application requests
- View and edit high level information pertaining to projects
- Manage partnering hospital access to the application
- Send out notifications to users

3.1.2.1.1 Receive and Process Application Requests
The console must allow RSF staff access to submitted applications, where they can process the information and sync new applicant data to NeonCRM.

3.1.2.1.2 View and Edit High Level Project Information
RSF staff shall be able to view high level project information such as:

- Calendar of Events
- Overview of past, current, and upcoming projects
- Financial Reports for each project

3.1.2.1.3 Manage partnering hospital access to the application
RSF staff shall be able to view and edit records of partnering hospitals. Staff will be able to add/remove hospitals and hospital staff from the system. RSF staff will also need access to shipping information for sending ready bags.

3.1.2.1.4 Send notifications to users

RSF staff shall be able to send out notifications to users or groups of users. These can be generated based on an event or customized by staff members.

3.1.2.2 NeonCRM Data Synchronization (Bartosz Maj, James Rice)

The system shall provide the ability to exchange all the data with Roc Solid Foundation’s NeonCRM system. The sending and receiving of data shall be done through secure methods.

3.1.2.2.1 RocStar shall use the API provided by NeonCRM’s creators to exchange data.

The system shall contain objects that will translate the information between NeonCRM and RocStar Database when requested by a user with security privileges of the administrator level. The following data objects shall be available to be synchronized upon user request:

- User
- Address
- Referral form
- Project
- Hospital

3.1.2.3 Data Storage and Administration (Bartosz Maj)

RocStar shall rely on data stored in a relational database management system, and must provide secure and reliable access to all the stored information. The database shall contain 17 tables.

3.1.2.3.1 RocStar shall use MySQL database to store all the necessary information.

Each database table shall meet the following functional requirements:

3.1.2.3.1.1 User Table
The User table shall hold the information of the application’s registered users. Each entry shall represent a single individual user. The following functional requirements shall be met:

a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique
c. The attribute Group shall be a foreign key referencing an entry in the UserGroup table.
d. The attribute Address shall be a foreign key referencing an entry in the Address table.
e. The attribute EmployedByHospital shall be a foreign key referencing an entry in the Hospital table.
f. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>GUID</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>INT</td>
<td>N</td>
<td>Foreign Key</td>
<td>Group that user belongs to</td>
</tr>
<tr>
<td>FirstName</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastName</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verified</td>
<td>BOOLEAN</td>
<td>Y</td>
<td></td>
<td>User verified email address</td>
</tr>
<tr>
<td>Email</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Addresss</td>
<td>GUID</td>
<td>N</td>
<td>Foreign Key</td>
<td></td>
</tr>
<tr>
<td>EmployedByHospital</td>
<td>GUID</td>
<td>N</td>
<td>Foreign Key</td>
<td>Non-null if user is Hospital Staff</td>
</tr>
</tbody>
</table>

Table 1: User Table Requirements

3.1.2.3.1.2 UserGroup Table
The UserGroup shall hold information describing user’s security access rights. The relationship between User and UserGroup is N:1, where one user can belong to only one group, but one group can contain multiple users.

a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique and shall auto-increment with every new entry.
c. The attribute AccessRights shall be a foreign key referencing an entry in the AccessRights table.
d. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>INT</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>AccessRights</td>
<td>INT</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: UserGroup Table Requirements

3.1.2.3.1.3 AccessRights Table
The AccessRights table shall hold information describing security access to different parts of the RocStar application.

a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique and shall auto-increment with every new entry.
c. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>INT</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>SecurityLevel</td>
<td>INT</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: AccessRights Table Requirements

3.1.2.3.1.4 FamilyPageAccess Table
The FamilyPageAccess shall hold information describing access to every existing Family Page for application’s users.

a. The attribute UserId shall be a foreign key referencing an entry in the User table.
b. The attribute FamilyPageld shall be a foreign key referencing an entry in the FamilyPageld table.
c. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserId</td>
<td>GUID</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
<tr>
<td>FamilyPageId</td>
<td>GUID</td>
<td>Y</td>
<td>ForeignKey</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: FamilyPageAccess Table Requirements

3.1.2.3.1.5 FamilyPage Table
The FamilyPage shall hold information describing access to every existing Family Page for application’s users.

a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique.
c. The attribute CreatedBy shall be a foreign key referencing an entry in the User table.
d. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>GUID</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>ChildName</td>
<td>VARCHAR(MAX)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CreatedBy</td>
<td>GUID</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
<tr>
<td>CreatedOn</td>
<td>DATE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: FamilyPage Table Requirements

3.1.2.3.1.6 ProjectAssignment Table
The ProjectAssignment table shall hold information regarding user’s role in an existing project

a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique and shall auto-increment with every new entry.
c. The attribute UserId shall be a foreign key referencing an entry in the User table.
d. The attribute ProjectId shall be a foreign key referencing an entry in the Project table.
e. The attribute ProjectRole shall be a foreign key referencing an entry in the ProjectRole table.
f. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>INT</td>
<td>Y</td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>UserId</td>
<td>GUID</td>
<td>Y</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>ProjectId</td>
<td>GUID</td>
<td>Y</td>
<td></td>
<td>Foreign Key</td>
</tr>
<tr>
<td>ProjectRole</td>
<td>INT</td>
<td>Y</td>
<td></td>
<td>Foreign Key</td>
</tr>
</tbody>
</table>

Table 6: ProjectAssignment Table Requirements

3.1.2.3.1.7 ProjectRole Table
The ProjectRole table shall hold information regarding possible roles that can be assigned to a user participating in a project.
  a. The attribute Id shall be the table’s primary key.
  b. The attribute Id shall be unique and shall auto-increment with every new entry.
  c. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>INT</td>
<td>Y</td>
<td></td>
<td>Primary Key</td>
</tr>
<tr>
<td>Name</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7: ProjectRole Table Requirements

3.1.2.3.1.8 Address Table
The Address table shall hold information regarding an existing physical address.
  a. The attribute Id shall be the table’s primary key.
  b. The attribute Id shall be unique.
  c. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>GUID</td>
<td>Y</td>
<td></td>
<td>Primary Key</td>
</tr>
</tbody>
</table>
### 3.1.2.3.1.9 Referral Table

The Referral table shall hold information regarding a patient referral created by an existing user.

- **a.** The attribute `Id` shall be the table’s primary key.
- **b.** The attribute `Id` shall be unique and shall auto-increment with every new entry.
- **c.** The attribute `ReferringUserId` shall be a foreign key referencing an entry in the User table.
- **d.** The attribute `Status` shall be a foreign key referencing an entry in the Status table.
- **e.** The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>GUID</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>ReferringUserId</td>
<td>GUID</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
<tr>
<td>PatientName</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReceivedOn</td>
<td>DATE</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>INT</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
</tbody>
</table>

### 3.1.2.3.1.10 Project Table

The Project table shall hold information regarding an RSF project created by an existing user.
a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique.
c. The attribute Status shall be a foreign key referencing an entry in the Status table.
d. The attribute Type shall be a foreign key referencing an entry in the ProjectType table.
e. The attribute Location shall be a foreign key referencing an entry in the Address table.
f. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>GUID</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>INT</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
<tr>
<td>StartDate</td>
<td>DATE</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EndDate</td>
<td>DATE</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>INT</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
<tr>
<td>TargetBudget</td>
<td>FLOAT(2)</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AvailableBudget</td>
<td>FLOAT(2)</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>GUID</td>
<td>N</td>
<td>Foreign Key</td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Project Table Requirements

3.1.2.3.1.11 Hospital Table
The Hospital table shall hold information regarding hospitals associated with RSF.
a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique.
c. The attribute Address shall be a foreign key referencing an entry in the Address table.
d. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
</table>

Table 10: Project Table Requirements
3.1.2.3.1.12 ProjectType Table
The ProjectType table shall enumerate different types of possible projects that can be created in the application.

a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique and shall auto-increment with every new entry.
c. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>INT</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 12: Project Table Requirements

3.1.2.3.1.13 Status Table
The Status table shall enumerate possible statuses that can be created in the application and assigned to existing projects and existing referrals.

a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique and shall auto-increment with every new entry.
c. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>INT</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 13: Project Table Requirements

3.1.X.1.1.14 ProjectInventory Table
The ProjectInventory table shall hold data regarding the required and available inventory for a created project.

a. The attribute ProjectId shall be a foreign key referencing an entry in the Project table.
b. The attribute ItemId shall be a foreign key referencing an entry in the Item table.
c. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjectId</td>
<td>GUID</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
<tr>
<td>ItemId</td>
<td>GUID</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
<tr>
<td>RequestedQuantity</td>
<td>FLOAT(2)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AvailableQuantity</td>
<td>FLOAT(2)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 14: ProjectInventory Table Requirements

3.1.2.3.1.15 Item Table
The Item table shall enumerate possible items that can be required as a part of a project.
a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique and shall auto-increment with every new entry.
c. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>INT</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 15: Item Table Requirements

3.1.2.3.1.16 ProjectTask Table
The ProjectInventory table shall hold data regarding the tasks required to complete a created project.
a. The attribute Id shall be the table’s primary key.
b. The attribute Id shall be unique.
c. The attribute ProjectId shall be a foreign key referencing an entry in the Project table.
d. The attributes that the table shall contain are as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Non-null (Y/N)</th>
<th>Key</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>GUID</td>
<td>Y</td>
<td>Primary Key</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>VARCHAR(MAX)</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProjectId</td>
<td>GUID</td>
<td>Y</td>
<td>Foreign Key</td>
<td></td>
</tr>
<tr>
<td>IsCompleted</td>
<td>BOOLEAN</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DueDate</td>
<td>DATE</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 16: Item Table Requirements

### 3.1.2.1 RocStar Notification System (Joshua Harris)

#### 3.1.2.1.1 Store Notifications

RocStar must be able to push notifications up to the server on the completion of certain actions.

#### 3.1.2.1.2 Send Notifications

RocStar must be able to send push notifications to users