
<Mercy Clinic>

<Volunteer Solution>
Use-Case 1: <Volunteer Schedules Appointment>

Version <1.1>

Revision History

Date	Version	Description	Author
10/13/2019	1.0	Initial Draft	Nestor Guerrero
10/25/2019	1.1	Revision of Main Success	Nestor Guerrero
10/25/2019	1.1	Completed Revision of Main Success	Kenzie Clarke

Use Case 1: Volunteer Schedules Appointment

Brief Description:

Mercy Clinic operates with the help of Volunteers, who offer medical, dental, or support services. It is essential to the operation of Mercy Clinic for a volunteer to schedule their volunteer shifts during Mercy's hours of operation. Instead of using the physical whiteboard in house at the Mercy Clinic, this feature allows the volunteer to schedule themselves by selecting an available appointment on the virtual calendar displayed by the system's front end. The Volunteer will also review the selection, provide verification for the system to prove they are the actual volunteer and not a stranger trying to fraud the schedule system, and then submits the request.

Primary Actor: Volunteer

Level: User Goal

Stakeholders and Interests: Volunteer (Medical, Dental, Support), Administrator, Owner of Mercy Clinic

Preconditions: None

Postconditions: Volunteer has filled in time slot for which they will be present to volunteer at the clinic

Trigger: Volunteer finds available time slot

Main Success Scenario:

1. Volunteer navigates to the schedule system webpage
2. System requests time slot information(number of time slots, day of slots, available or filled, shift time, shift type) for the current month from the Database
3. Database returns the time slot information to the System.
4. System displays the time slot information to the user.
5. Volunteer selects an available time slot from presented options.
6. System asks the user for their primary mode of contact (email or phone number)
7. System confirms with the Database that the entered mode of contact exists.
8. System send a 6-digit verification code to the mode of contact.
9. Volunteer enters 6-digit verification code.
10. System validates 6-digit verification code..
11. The System asks for confirmation of the selected time slot.
12. Volunteer confirms selected time slot.
13. System presents a message of confirmation with a path back to the calendar.

Extensions:

- 2a. Volunteer sees no available time slots for the current week.
 - 2a1. Volunteer will go to the next week to see later available slots.
- 3a. Volunteer sees no time slots they want.
 - 3a1. Volunteer exits system.
- 5a. Volunteer wants to change a date.
 - 5a1. Volunteer returns to schedule to view available time slots

Priority: High

<Mercy Clinic>

Secondary Actors: System, Database

Special Requirements: The Volunteer is only allowed to schedule their appointment so far in advance. This length is going to be specified by the client in our next meeting.

Open Issues: None

Mercy Clinic

Volunteer Solution
Use-Case 2: Volunteer Cancels Appointment

Version <1.2>

Revision History

Date	Version	Description	Author
10/11/2019	1.0	initial draft	Kenzie Clarke
10/21/2019	1.1	content updated after peer review gramatical errors corrected numbering of use case altered	Kenzie Clarke
10/25/19	1.2	switched order or verification and confirmation	Kenzie Clarke

Use Case 2: Volunteer Cancels an Appointment

Brief Description:

Within the scheduler system, there are two main functionalities for the volunteer actor - this includes the action of scheduling an appointment and canceling an appointment. If a volunteer wishes to modify an appointment, they must first cancel their appointment and then schedule a new appointment. This use case document details the case of the volunteer canceling their own previously scheduled appointment. This might happen either because the scheduling was done on accident or because the volunteer can no longer maintain that appointment due to unforeseen circumstances.

Primary Actor: Volunteer

Level: User goals

Stakeholders and Interests: Volunteer, Admin, Owner, other Volunteers, Patients

Preconditions: Volunteer has scheduled an appointment

Postconditions: Appointment is removed from the scheduler database

Appointment time becomes available in the scheduler display

Volunteer is sent a confirmation message of appointment deletion

Trigger: The volunteer has indicated they would like to cancel their appointment

Main Success Scenario:

1. The volunteer indicates that they would like to cancel a selected appointment.
2. The schedule system asks for and receives from the volunteer their primary mode of contact.
3. The schedule system verifies the volunteer primary mode of contact using the database system.
4. The schedule system generates and sends a verification code to the primary mode of contact.
5. The volunteer received the verification code on their primary mode of contact and enters it into the schedule system.
6. The schedule system confirms the verification code.
7. The schedule system presents the selected appointment details and asks for confirmation of deletion
8. The volunteer confirms that the selected appointment is the correct appointment.
9. The schedule system submits the deletion request to the database system.
10. The database system removes the row corresponding to the appointment.
11. The database system send confirmation of deletion to the schedule system.
12. The schedule system sends a confirmation notice to the primary mode of contact.
13. The volunteer received the confirmation notice on their primary mode of contact.
14. The volunteer exits the system.

Extensions:

- 1a. The volunteer fails to select an appointment to cancel
 - 1a1. The volunteer indicates that they would like to cancel an appointment.

- 1a2. The schedule system asks the user to select an appointment they would like to cancel.
- 1a3. The schedule system exits the use case.
- 2a. The volunteer enters the wrong verification code
 - 2a1. The volunteer indicates that they would like to cancel a selected appointment.
 - 2a2. The schedule system presents the selected appointment details and asks for confirmation of deletion.
 - 2a3. The volunteer confirms that the selected appointment is the correct appointment.
 - 2a4. The schedule system asks the database system for the volunteer primary mode of contact.
 - 2a5. The database system retrieves the volunteer primary mode of contact for the schedule system.
 - 2a6. The schedule system generates and sends a verification code to the primary mode of contact.
 - 2a7. The volunteer enters the wrong verification code.
 - 2a8. The schedule system informs the volunteer of an incorrect code.
 - 2a9. Steps 2a6 - 2a8 are repeated two more times.
 - 2a10. The schedule system informs the volunteer of their failure to verify their identity.
 - 2a11. The schedule system asks the database system for the email of the administrator.
 - 2a12. The database system provides the email of the administrator.
 - 2a13. The schedule system sends an email to the administrator detailing the attempted cancelation and failure to verify.
 - 2a14. The schedule system exits the use case.

Priority: Medium-High

Secondary Actors: Scheduler system, database system, administrator

Special Requirements:

*[A special requirement is typically a **nonfunctional** requirement that is specific to a use case, but is not easily or naturally specified in the text of the use case's event flow. Examples of special requirements include legal and regulatory requirements, application standards, and quality attributes of the system to be built including **usability, reliability, performance or supportability requirements**. Additionally, other requirements—such as operating systems and environments, compatibility requirements, and design constraints—should be captured in this section.]*

Open Issues:

Mercy Clinic

Volunteer Solution
Use-Case 3: Admin Schedules an Appointment for a
Volunteer

Version <1.0>

Revision History

Date	Version	Description	Author
10/14/2019	1.0	Initial draft	Wynn Pho
10/21/2019	1.1	Made changes Dr. Wei suggested Updated document to make it more detailed	Wynn Pho

User Case 3: Admin Schedules an Appointment for a Volunteer

Brief Description:

This use case describes the case of an admin scheduling an appointment for a volunteer. The volunteer may ask the admin by either calling or meeting in person to help them schedule a volunteer slot. The admin gives the volunteer available time slots for the volunteer to pick one, then the admin puts it in the scheduler system.

Primary Actor: Admin

Level: User goal

Stakeholders and Interests: Volunteer, Admin, Owner, other Volunteers, Patients

Preconditions: The admin knows the available time slot the volunteer wants. The admin is logged on to the system.

Postconditions: The appointment is scheduled. The scheduled time slot becomes unavailable on the website. The confirmation text is sent to the volunteer.

Trigger: The volunteer has indicated they want to schedule an appointment.

Main Success Scenario:

1. Admin requests to see available time slots
2. The schedule system retrieves available time slots from the database and displays them to admin
3. Admin selects the time slot the volunteer wants scheduled
4. Admin puts in volunteer's name and contact number for that time slot
5. The time slot becomes unavailable. The scheduling table is updated
6. The schedule system sends a confirmation text to volunteer
7. Admin logs out

Extensions: None

Priority: High

Secondary Actors: Database system, Schedule system, Administrator

Special Requirements: None

*[A special requirement is typically a **nonfunctional** requirement that is specific to a use case, but is not easily or naturally specified in the text of the use case's event flow. Examples of special requirements include legal and regulatory requirements, application standards, and quality attributes of the system to be built including **usability, reliability, performance or supportability requirements**. Additionally, other requirements—such as operating systems and environments, compatibility requirements, and design constraints—should be captured in this section.]*

Open Issues:

1. How long is a time slot? 30 minutes? 1 hour? 2 hours?
2. What if multiple volunteers are needed at one time slot?
3. How does Admin put available time slots in the calendar? Do we need another use case for this?
4. Does the calendar display only current week's available time slots? Or does it also display next week's, the week after next week's, etc.?

Mercy Clinic

Volunteer Solution
Use-Case 4: Admin Cancels Appointment

Version <1.0>

Revision History

Date	Version	Description	Author
10/11/2019	1.0	initial draft	Kenzie Clarke

Use Case 4: Admin Cancels Appointment

Brief Description:

Within the scheduler system, there are many functionalities for the admin actor. This use case document details the case of the admin canceling a volunteer's previously scheduled appointment. This might happen either because the scheduling was done on accident or because the volunteer can no longer maintain that appointment due to unforeseen circumstances, and they have handed over the responsibility of updating the schedule to the admin.

Primary Actor: Admin

Level: User goals

Stakeholders and Interests: Volunteer, Admin, Owner, other Volunteers, Patients

Preconditions: Volunteer has scheduled an appointment

The admin is logged on to the system.

Postconditions: Appointment is removed from the scheduler database.

Appointment time becomes available in the scheduler display.

Volunteer is sent confirmation message of appointment deletion.

Trigger: The volunteer has indicated they would like to cancel their appointment.

Main Success Scenario:

1. The admin indicates that they would like to cancel a selected appointment.
2. The schedule system presents the selected appointment details and asks for confirmation of deletion.
3. The admin confirms that the selected appointment is the correct appointment.
4. The schedule system confirms the identity of the admin account.
5. The schedule system submits the deletion request to the database system.
6. The database system removes the row corresponding to the appointment.
7. The database system send confirmation of deletion to the schedule system.
8. The schedule system asks the database system for the appointment's volunteer primary mode of contact.
9. The database system retrieves the volunteer primary mode of contact for the schedule system.
10. The schedule system sends a confirmation notice to the primary mode of contact.
11. The volunteer receives the confirmation notice on their primary mode of contact.
12. The schedule system informs the admin of the successful deletion.
13. The admin exits the system.

Extensions:

- 1a. The admin fails to select an appointment to cancel
 - 1a1. The admin indicates that they would like to cancel an appointment.
 - 1a2. The schedule system asks the admin to select an appointment they would like to cancel.

- 1a3. The schedule system exits the use case.
- 2a. The database system is offline or fails to communicate with the schedule system
 - 2a1. The admin indicates that they would like to cancel a selected appointment.
 - 2a2. The schedule system presents the selected appointment details and asks for confirmation of deletion
 - 2a3. The admin confirms that the selected appointment is the correct appointment.
 - 2a4. The schedule system confirms the identity of the admin account.
 - 2a5. The schedule system submits the deletion request to the database system.
 - 2a6. The schedule system times out without receiving information from the database system.
 - 2a7. The schedule system informs the admin of the database error.
 - 2a7. The schedule system exits the system.

Priority: Medium-High

Secondary Actors: Scheduler system, database system, administrator

Special Requirements:

Open Issues:

Mercy Clinic

**Volunteer Solution
Use-Case 5: Volunteer Checks In**

Version <1.1>

Revision History

Date	Version	Description	Author
10/13/19	1.0	Initial draft	Cole Weber
10/20/19	1.1	Updated to reflect received feedback and comments	Cole Weber
11/6/19	1.1	Cleaned up document	Warren Riley

UC05 Volunteer Checks In

Brief Description:

This use case covers the role of a volunteer checking into the clinic. It describes a volunteer using the check-in system and how the system processes the check-in.

Primary Actor: Volunteer

Level: User goal

Stakeholders and Interests: Volunteers, Administrators

Preconditions: Volunteer has been approved by clinic and is entered in database system

Postconditions: System logs check in time into to database

Trigger: Volunteer arrives at clinic to begin shift

Main Success Scenario:

1. Volunteer indicates that they would like to check in and enters ID (their email address) into check in system device
2. Check in system verifies with database system that volunteer exists and that they are not already checked in
3. Check in system confirms with volunteer that they desire to check in at the present time
4. Volunteer confirms check in
5. Check in system tells database system to record check in of volunteer at desired time and date
6. System indicates check-in is done with confirmation message

Extensions:

- 1a. Volunteer is not in system or enters wrong ID
 - 1a1. Check in system is unable to verify with database system that volunteer exists
 - 1a2. Check in system alerts user that ID is invalid or does not exist
 - 1a3. Volunteer enters ID again if necessary
- 2a. Volunteer forgets to check in
 - 2a1. Volunteer notifies Administrator of missed clock in
 - 2a2. Administrator logs into database system and manually records shift session for volunteer (performs record shift session use case)
- 3a. Volunteer is already checked in
 - 3a1. Check in system detects volunteer is already checked in
 - 3a1. Check in system alerts user that they are already checked in and directs them to check out
 - 3a2. Volunteer performs check out use case if necessary

Priority: High

Mercy Clinic

Secondary Actors: Check in / out system, database system, administrator

Special Requirements:

Open Issues:

Should the check in system also verify that a volunteer is on the schedule or should anyone be able to check in if they arrive? Might have sign next to check in system that tells volunteers not on schedule to check with Lorena first before checking in, so she can add them to the schedule manually or have her check in the volunteer in herself. Don't know the clinic's policy on turning away extra volunteers, so will also need to find this out.

Mercy Clinic

**Volunteer Solution
Use-Case 6: Volunteer Checks Out**

Version <1.1>

Revision History

Date	Version	Description	Author
10/13/19	1.0	Initial draft	Cole Weber
10/20/19	1.1	Updated to reflect received feedback and comments	Cole Weber
11/6/19	1.1	Cleaned up document	Warren Riley

UC06 Volunteer Check Out

Brief Description:

This use case covers the role of a volunteer checking out of the clinic. It describes a volunteer using the check-out system and how the system processes the check-out.

Primary Actor: Volunteer

Level: User goal

Stakeholders and Interests: Volunteers, Administrators

Preconditions: Volunteer has already checked into system

Postconditions: System logs check out time and calculates hours worked

Trigger: Volunteer finishes shift at clinic

Main Success Scenario:

1. Volunteer indicates that they would like to check out and enters ID (their email address) into check out system device
2. Check out system verifies with database system that volunteer exists and that they are checked in
3. Check out system confirms with volunteer that they desire to check out at present time
4. Volunteer confirms check out
5. Check out system tells database system to record check out of volunteer at desired time and date
6. Database system calculates hours worked during shift
7. System indicates check-out is done with confirmation message

Extensions:

- 1a. Volunteer is not in system or enters wrong ID
 - 1a1. Check out system is unable to verify with database system that volunteer exists
 - 1a2. Check out system alerts user that ID is invalid or does not exist
 - 1a3. Volunteer enters ID again if necessary
- 2a. Volunteer forgets to check out
 - 2a1. Volunteer notifies Administrator of missed clock out
 - 2a2. Administrator logs into database system and manually updates check out for volunteer (performs update shift session use case)
 - 2b0.. Possible feature to implement...
 - 2b1. Every night at midnight on days clinic is open, check in system looks for volunteers who did not check out
 - 2b2. Check in system emails administrator(s) (most likely Lorena) a list of all volunteers who did not check out
 - 2b3. Administrator logs into database system and manually updates check out for volunteers on list (performs update shift session use case)
- 3a. Volunteer is already checked out

Mercy Clinic

- 3a1. Check out system determines from database system that volunteer is already checked out
- 3a2. Check out system alerts user that they are already checked out and directs them to check in
- 3a3. Volunteer performs check in use case if necessary

Priority: High

Secondary Actors: Check in / out system, database system, administrators

Special Requirements:

Open Issues:

Mercy Clinic

Volunteer Solution
Use-Case 7: Admin Generates Hours Report For One
Volunteer

Version <1.0>

Revision History

Date	Version	Description	Author
10/16/2019	1.0	Initial Draft	Minh Nguyen

Use Case 7: Admin Generates Hours Report For One Volunteer

Brief Description:

This use case describes the case of the admin generates hours report for one volunteer. The volunteer might request the admin to generate an hours report for educational records or social credit hours. With database containing the hours that the volunteer had worked, the admin will generate the hours report that satisfies the time frame based on the volunteer's request. The report is an excel file that can be either printed out or emailed to the volunteer depending on which type of report the volunteer wants.

Primary Actor: Admin

Level: User goals

Stakeholders and Interests: Volunteer, Admin, Non-profit Organization

Preconditions: Volunteer has completed the appointments with check in / check out system
The admin is logged on to the system

Postconditions: The hours report will be generated upon the request
The volunteer will receive the report electronically or on paper

Trigger: The volunteer has requested the admin to generate the hours report or the admin would like to know the numbers of hours completed by a volunteer

Main Success Scenario:

1. The volunteer requests the admin to generate an hourly report via phone, email or face to face interaction
2. The admin accepts the request, responds to the volunteer and logs on to the system
3. The admin selects the volunteer's name from database
4. The admin checks with the volunteer about phone number, email and birthday stored in the database to make sure that they match
5. The admin asked the volunteer for the desired time frame that the volunteer wants on the report
6. The admin chooses to see the hours database of the volunteer and the database displays it to the admin
7. The admin selects the time frame upon the volunteer requests
8. The admin generates the hours report for the volunteer
9. The volunteer receives the report via email or on paper
10. The admin logs out

Extensions:

- 3a. The admin does not find the volunteer information in the database
 - 3a1. The volunteer requests the admin to generate an hourly report via phone, email or face to face interaction
 - 3a2. The admin accepts the request, responds to the volunteer and logs on to the system
 - 3a3. The admin selects the volunteer's name from database
 - 3a4. The admin checks with the volunteer through other information stored in the database: phone number, emails, and birthday
 - 3a5. The admin rejects to generate the report for the volunteer if other information are not verified
 - 3a6. The admin logs out

Mercy Clinic

- 6a. The time frame that the volunteer requests is not available
 - 6a1. The volunteer requests the admin to generate an hourly report via phone, email or face to face interaction
 - 6a2. The admin accepts the request, responds to the volunteer and logs on to the system
 - 6a3. The admin selects the volunteer's name from database
 - 6a4. The admin checks with the volunteer about phone number, email and birthday stored in the database to make sure that they match
 - 6a5. The admin chooses to see the hours database of the volunteer and the database displays it to the admin
 - 6a6. The admin discovered that the time frame upon the volunteer requests is not available
 - 6a7. The admin rejects to generate the report for the volunteer
 - 6a8. The admin logs out

Priority: High

Secondary Actors: scheduler system, check in / check out system, database system, administrator

Special Requirements: searching criteria

Open Issues:

Mercy Clinic

Volunteer Solution
Use-Case 8: Admin Generates Hours Report Of All
Volunteers

Version <1.0>

Revision History

Date	Version	Description	Author
10/16/2019	1.0	Initial Draft	Minh Nguyen

Use Case 8: Admin Generates Hours Report Of All Volunteers

Brief Description:

This use case describes the case of the admin generates hours report of all volunteers. The owners and the donors might request an hourly report of all the volunteers from the admin annually. The purpose of this report is to request funding from donors and owners of the clinic. From the hours report of all the volunteers, the donors and owners can see how the clinic is operated annually and will decide how much funding to donate.

Primary Actor: Admin

Level: User goals

Stakeholders and Interests: Volunteers, Admin, Owners, Donors

Preconditions: Volunteers have scheduled appointments

Volunteers have completed the scheduled appointments with check in / check out system

The records are saved into a database

The admin is logged on to the system and has administrative access to database of all the volunteers

Postconditions: The hours report will be generated upon the request of the owners and the donors

The owners and the donors will receive the report electronically and paper

The owners and the donors will decide how much funding to donate to the clinic

Trigger: The owners and the donors have requested the admin to generate the hours report

Main Success Scenario:

1. The owners and the donors request the admin to generate an hourly report of all the volunteers
2. The admin accepts the request, responds to both donors and owners and logs on to the system
3. The admin selects all volunteer's name from database based on the criteria specified by the owners and donors
4. The admin chooses to see the all the hours of all the volunteers in the database
5. The admin selects the time frame upon the owners and donors requests
6. The admin generates the hours report of all the volunteers for the owners and donors
7. The owners and donors receive the report via email and on paper
8. The admin logs out

Extensions: None

Priority: Medium-High

Secondary Actors: scheduler system, check in / check out system, database system

Special Requirements: search criteria

Open Issues:

Mercy Clinic

Volunteer Solution
Use-Case 9: User Select Appointment

Version <1.3>

Revision History

Date	Version	Description	Author
10/21/19	1.0	Initial draft	Cole Weber
10/25/19	1.3	Updated Main Success Senario Added Extensions 2a and 6a Updated Special Requirements Updated Preconditions	Kenzie Clarke

UC09 User Selects Appointment

Brief Description:

This use case covers the role of a user (volunteer or administrator) using the scheduling system to select an appointment. This is an important ability for the users as it is the first step in performing tasks such as scheduling and appointment or canceling one.

Primary Actor: User**Level:** User Goal**Stakeholders and Interests:** Volunteer, Administrator**Preconditions:** None**Postconditions:** Desired appointment is selected**Trigger:** User wishes to interact (schedule or cancel) with an appointment**Main Success Scenario:**

1. User navigates to appointments calendar page in scheduling system
2. Scheduling system retrieves appointment data (days in month, slots per day, slot information (position title, shift start and end time, date), and availability status of each slot) from the Database.
3. Scheduling system displays dates and number of slots available for each day on the calendar.
4. User selects day they would like to view.
5. Scheduling system displays all appointments information(slot number, position title, start time, end time, availability status) for that day.
6. User selects the slot they wish to interact with.

Extensions:

- 2a. The Database returns an error.
 2. Scheduling system retrieves an error from the Database.
 3. Scheduling system informs the user of the error and asks the user to refresh the system.
- 6a. There are no available slots.
 6. User navigates back to the calendar.

Priority: High**Secondary Actors:** Scheduling system, Database**Special Requirements:** The User needs access to the internet to refresh the page (see extension 2a).**Open Issues:**

Mercy Clinic

Volunteer Solution
Use-Case 10: Admin Records Shift Session

Version <1.0>

Revision History

Date	Version	Description	Author
10/21/19	1.0	Initial draft	Cole Weber
11/6/18	1.0	Cleaned up document	Warren Riley

UC10 Admin Records Shift Session

Brief Description:

This use case covers the role of an Administrator manually recording a shift session for a volunteer. A volunteer may forget to check in and needs their shift recorded. In this case, an Admin will log into the database system and manually record a shift session for the volunteer.

Primary Actor: Administrator**Level:** User Goal**Stakeholders and Interests:** Volunteer, Administrator**Preconditions:** Admin is logged into the database system**Postconditions:** Volunteer shift session is recorded in database**Trigger:** Admin is notified by volunteer that they need a shift session recorded**Main Success Scenario:**

1. Admin indicates that they would like to record a new shift session
2. Database system asks Admin for ID (email address) of volunteer and verifies that they exist
3. Database system ask Admin to enter date, time in and time out for volunteer
4. Admin fills in requested information
5. Database confirms that shift session has been recorded for volunteer (possibly sends user an email confirmation as well)

Extensions:

- 1a. Volunteer is not in system or wrong ID is entered
 - 1a1. Database system is unable to verify that volunteer exists
 - 1a2. Database system alerts Admin that ID is invalid or does not exist
 - 1a3. Admin enters ID again if necessary

Priority: High**Secondary Actors:** Database system, Volunteer**Special Requirements:****Open Issues:**

Mercy Clinic

Volunteer Solution
Use-Case 11: Admin Updates Shift Session

Version <1.0>

Revision History

Date	Version	Description	Author
10/21/19	1.0	Initial draft	Cole Weber

UC11 Admin Updates Shift Session

Brief Description:

[The description briefly conveys the role and purpose of the use case. A single paragraph will suffice for this description.]

This use case covers the role of an Administrator manually updating a shift session for a volunteer. A volunteer may forget to check out needs their shift updated. In this case, an Admin will log into the database system and manually correct the volunteer's mistake.

Primary Actor: Administrator

Level: User Goal

Stakeholders and Interests: Volunteer, Administrator

Preconditions: Admin is logged into the database system

Postconditions: Volunteer shift session is corrected in database

Trigger: Admin is notified by volunteer (or potentially check in / out system) that they did not check out or need time adjusted

Main Success Scenario:

[This use case starts when the actor does something. An actor always initiates use cases. The use case describes what the actor does and what the system does in response. It is phrased in the form of a dialog between the actor and the system.]

The use case describes what happens inside the system, but not how or why. If information is exchanged, be specific about what is passed back and forth. For example, it is not very illuminating to say that the actor enters customer information. It is better to say the actor enters the customer's name and address. A Glossary of Terms is often useful to keep the complexity of the use case manageable, you may want to define things like customer information there to keep the use case from drowning in details.

*Alternatives may be presented within the text of the use case. If it only takes a few sentences to describe what happens when there is an alternative, use a separate section to describe it. For example, an **Extensions** subsection explains how to describe alternatives.*

A picture is sometimes worth a thousand words, though there is no substitute for clean, clear prose. If it improves clarity, feel free to paste graphical depictions of user interfaces, process flows or other figures into the use case. If a flow chart is useful to present a complex decision process, by all means use it! Similarly for state-dependent behavior, a state-transition diagram often clarifies the behavior of a system better than pages upon pages of text. Use the right presentation medium for your problem, but be wary of using terminology, notations or figures that your audience may not understand. Remember that your purpose is to clarify, not obscure.]

1. Admin indicates that they need to correct a volunteer shift session
2. Database system asks Admin for ID (email address) of volunteer and verifies that they exist
3. Database system lists shift sessions for selected volunteer
4. Admin selects session to correct
5. Database system allows admin to modify time of session
6. Admin corrects shift session
7. Database confirms that shift session has been updated for volunteer (possibly sends user an email confirmation as well)

Extensions:

- 1a. Volunteer is not in system or wrong ID is entered
 - 1a1. Database system is unable to verify that volunteer exists

1a2. Database system alerts Admin that ID is invalid or does not exist

1a3. Admin enters ID again if necessary

Priority: High

Secondary Actors: Volunteer, Database system

Special Requirements:

*[A special requirement is typically a **nonfunctional** requirement that is specific to a use case, but is not easily or naturally specified in the text of the use case's event flow. Examples of special requirements include legal and regulatory requirements, application standards, and quality attributes of the system to be built including **usability, reliability, performance or supportability requirements**. Additionally, other requirements—such as operating systems and environments, compatibility requirements, and design constraints—should be captured in this section.]*

Open Issues:

Mercy Clinic

**Volunteer Solution
Use-Case 12: Admin Logs In**

Version <1.0>

Revision History

Date	Version	Description	Author
10/25/19	1.0	Initial draft	Kenzie Clarke

UC12: Admin Logs In

Brief Description:

For the Mercy Clinic Volunteer Solution Webpage, there are accounts created for administrators. An administrator must first verify themselves by logging in to their account on the webpage. This use case details the process of an admin user logging into their admin account.

Primary Actor: Admin user

Level: User Goal

Stakeholders and Interests: Volunteer, Administrators

Preconditions: None

Postconditions: Admin user is logged into the webpage

Trigger: Admin user asks the system for administrator access to the webpage.

Main Success Scenario:

1. Admin user asks the system to log in to their admin account.
2. The system asks for and receives from the admin user a username and password associated with the admin user's account.
3. The system asks the database to verify the username and password.
4. The database verifies the account information.
5. The system updates the webpage to show the admin user logged in and removes verification requirements for admin user.

Extensions:

- 4a. Admin user is not verified by the database.
 - 4a1. The database has no entry that matches the username.
 4. Database returns username error to the system.
 5. The system informs the user of the non-existent username.
 6. return to step 2.
 - 4a2. The database cannot verify the provided password.
 4. Database returns password error to the system.
 5. The system informs the user of the incorrect password.
 6. Repeat steps 2-6 two more times.
 7. The system retrieves the email of administrator and sends an alert concerning the failed login attempt.
 8. The system informs the admin user of the failure to enter the correct password.
 9. The system exits the use case.

Priority: High

Secondary Actors: System, Database

Special Requirements: The webpage will prevent the admin user from entering a password more than three times in a row. It will warn the user about a time out feature on the second failure. If the user fails to correctly input their password after three attempts - they will be locked out of the system for 5 minutes.

Open Issues:

Mercy Clinic

Volunteer Solution
Use-Case 13: Admin Creates Schedule

Version <1.2>

Revision History

Date	Version	Description	Author
05/11/2019	1.0	Initial Draft	Minh Nguyen
11/11/2019	1.2	Changed Title	Kenzie Clarke

UC 13: Admin Creates Schedule

Brief Description:

Admin have the administrative access to the system as only admin holds an account in the system. For the Volunteers to schedule or cancel their shifts, admin must create a schedule by putting empty time slots with the job function on a specific date in a calendar that requires the job.

Primary Actor: Admin

Level: Users Goal

Stakeholders and Interests: Admin, Volunteers, Owner

Preconditions: Admin is logged into to system through the administrative account

Postconditions: Admin add time slots to the calendar

Volunteers are able to schedule for their shift in advance

Trigger: A schedule is needed for next month

More volunteers are needed for a job, but there is no available slots on the current schedule for the volunteers to sign up

Main Success Scenario:

1. Admin chooses a month that has not been scheduled to schedule
2. Admin selects a date in that month
3. Admin adds the available time slots with the job function on that date
4. The new time slots are added into the database
5. The system displays available time slots with job functions of that date for volunteers to sign up
6. Admin repeats the steps 2-3 for other dates in the month
7. Admin logs out of the system

Extensions:

- 1a. Admin chooses a month that has been scheduled
 - 1a1. Admin selects a date in that month that requires more volunteers than the number of slots the admin has previously scheduled
 - 1a2. Admin adds the available time slots with the job function on that date
 - 1a3. The new time slots are added into the database
 - 1a4. The system displays new open time slots with job function for volunteer to sign up
 - 1a5. Return to the step 5 in the main scenario
- 2a. Admin selects a date in the month with occurrence options
 - 2a1. Admin select occurrence options of daily, weekly, or monthly based on the need of the jobs for the clinic
 - 2a2. Admin adds the available time slots with the job function on that date and its occurrence
 - 2a2a. Admin edits one time slot in an occurrence without changing the rest of the occurrences

- 2a2a1. Admin selects a date that has time slots in an occurrence
- 2a2a2. Admin selects one time slot to edit
- 2a2a3. The system asks the admin to edit the whole occurrence starting from that date or just this particular time slot
- 2a2a4. Admin select option to edit one time slot in the occurrence
- 2a2a5. Admin edits it based on the need of the job function on that date
- 2a2a6. The time slots are edited in database
- 2a2a7. The system displays the new edited time slots but the rest in the occurrence remains the same
- 2a2a8. Admin repeats step 2a2a1 to 2a2a4
- 2a2a9. Return to step 5 in the main success scenario
- 2a2b. Admin edits all of the time slots in the occurrences
 - 2a2b1. Admin selects a date that has time slots in an occurrence
 - 2a2b2. Admin selects one time slot to edit
 - 2a2b3. The system asks the admin to edit the whole occurrence starting from that date or just this particular time slot
 - 2a2b4. Admin select option to edit every time slots in the occurrence
 - 2a2b5. Admin edits it based on the need of the job function
 - 2a2b6. The time slots are edited in database
 - 2a2b7. The system displays the new edited time slots in an occurrence starting from the date that the admin chooses
 - 2a2b8. Admin repeats step 2a2a1 to 2a2a4
 - 2a2b9.. Return to step 5 in the main success scenario
- 2a3. The time slots are added into the database
- 2a4.. The system displays available time slots with job functions on that date and its occurrence for volunteers to sign up
- 2a5. Return to the step 5 in the main success scenario

Priority: High

Secondary Actors: System, database

Special Requirement: Occurrence option when creating a schedule from administrative account

Open Issues:

< Mercy Clinic >

<Volunteer Solution >
Use-Case <14>: <Admin Forgets Password >

Version <1.0 >

Revision History

Date	Version	Description	Author
11/1/19	1.0	Initial Use Case Main success scenario Extensions 5a and 9a	Kenzie Clarke

UC14: Admin Forgets Password

Brief Description:

In this system, only an administrator holds an account. This account is kept secure with a password, which the user must input along with a email/username to log in to the system as an administrator. In the event that the user forgets the password associated with their administrator account, this use case allows them to recover that password.

Primary Actor: Administrator

Level: User Goals

Stakeholders and Interests: Administrators, Owner

Preconditions: Administrator is not logged in to the system.

Postconditions: Administrator is able to log in to the system with their email/username and recovered password.

Trigger: Administrator has informed the system that they have forgotten their password

Main Success Scenario:

1. Administrator has informed the system that they have forgotten their password.
2. The system asks the Admin to input their email/username.
3. The Admin gives and confirms their email/username.
4. The system asks the database for validation of the email/username.
5. The database confirms the existence of the email/username in the tables holding administrator account information.
6. The system asks the database for the recovery email and the password of the administrator.
7. The database returns the recovery email and password of the administrator.
8. The system asks the Admin for confirmation of the recovery email.
9. The Admin confirms that they have access to the recovery email to the system.
10. The system sends an email containing the administrator's password.
11. The administrator recovers their password from the email.
12. The administrator proceeds to execute Use Case 12: Admin Logs In.

Extensions:

- 5a. The Admin inputs an incorrect username/email.
 - 5a1. The database denies the existence of the email/username in the tables holding administrator account information.
 - 5a2. The system informs the Admin of the denial.
 - 5a3. Return to step 4 of the main success scenario.
- 9a. Admin does not have access to their recovery email.
 - 9a1. Admin denies that they have access to their recovery email to the system.
 - 9a2. The system informs the Admin that they must contact the primary Administrator for assistance.

9a3. The system asks the database for the email of the primary Administrator

9a4. The database returns the email of the primary Administrator to the system.

9a5. The system send the primary Administrator an email containing information about the unsuccessful password recovery and required assistance.

9a6. The system exits (gracefully).

Priority: High

Secondary Actors: System, Database

Special Requirements:

To recover the password of the account, it follows that the administrator attempting to recover their password have actually have an account first. If they do not, then an account must be created for them by a current administrator.

Open Issues:

Mercy Clinic

Volunteer Solution
Use-Case 15: Admin Creates An Admin Account

Version <1.0>

Revision History

Date	Version	Description	Author
05/11/2019	1.0	Initial Draft	Minh Nguyen

UC 15: Admin Creates An Admin Account

Brief Description:

Currently, there is only one administrative account for one admin to access. However, in the event that Mercy Clinic decides that job needs more admins to share the load or the owners want to have administrative access to database, this use case can solve it by allowing the current admin to create an admin account

Primary Actor: Administrator

Level: User goal

Stakeholders and Interests: Administrator, owner, other administrators

Preconditions: Admin is logged in to the system

Admin has all the information needed of the new admins: name, gender, birthdate, address, email, phone to create an account

Postconditions: Admin successfully create an admin account

Trigger: Admin is requested to create an admin account

Main Success Scenario:

1. Admin requests the system to create a new admin account
2. The system asks the admin the password
3. The system confirms that is it the admin
4. The system displays a new page for the admin to create an admin account
5. The admin enter information into the required fields to create a new account: name, gender, birthdate, address, email and phone
6. The admin asks the new admin to input desired username and password
7. The admin signal the system to create an account
8. The system perform a checks of all the entered fields for an incorrect format and length
9. The system signal a new account has been created and an email verification will be sent to the new admin
10. The new admin confirms the verification through the email within 24 hours
11. The new admin information will be added to the database
12. The new admin can now have the same administrative access as the the current admin
13. Admin repeats from step 1 for other new admins
14. Admin log off the system

Extensions:

- 2a. Admin enters the wrong password
 - 2a1. The system signal error of wrong password
 - 2a2. Repeat until correct password is entered
- 6a. The new admin is not present
 - 6a1. The current admin enter a temporary username and password
 - 6a2. Repeat step through step 12

6a3. The new admin logs in and have option to change the username and password

6a4. The new information of username and password will be updated in the database

10a. The new admin does not confirm the verification through email within 24 hours

10a1. The new information of new admin account is not added to the database

10a2. Repeat from step 1

Priority: Medium

Secondary Actors: System, database

Special Requirements:

N/A

Open Issues:

Mercy Clinic

Volunteer Solution
Use-Case 16: Admin Logs Out

Version <1.0>

Revision History

Date	Version	Description	Author
11/05/19	1.0	Initial draft	Kenzie Clarke

UC12: Admin Logs Out

Brief Description:

For the Mercy Clinic Volunteer Solution Webpage, there are accounts created for administrators. An administrator must first verify themselves by logging in to their account on the webpage. This use case details the process of an admin user logging out of their admin account.

Primary Actor: Admin user

Level: User Goal

Stakeholders and Interests: Volunteer, Administrators

Preconditions: Admin user is logged into the webpage.

Postconditions: Admin user is not logged into the webpage.

Trigger: Admin user asks the system to exit their administrator account on the webpage.

Main Success Scenario:

1. Admin user asks the system to log out of their admin account.
2. The system updates the webpage to show no administrator logged in and adds verification requirements for admin user.

Extensions:

- 1a. Admin user is unable to ask the system to log out of their admin account
 2. Admin user contacts maintenance for assistance.
 3. Exits the use case.

Priority: High

Secondary Actors: System, Database

Special Requirements: None

Open Issues: