

U.R.S.

Only *Devs;*

Content

1. Description
2. Functional & Non-functional Requirements
3. Use Cases
4. User Stories
5. Wireframe
6. Version table

1. Description

Our goal in this project is to solve a problem that SIOUX put forward as part of a software engineering course at Fontys University of Applied Sciences. The project involves managing appointments visitors schedule with SIOUX employees and keeping track of parking lot capacity so visitors are notified if there is space to park in the parking lot.

The following document is intended to outline the functionalities of the app, the use cases and user stories of our software solution.

2. Functional & Non-functional requirements

Non-functional requirements

- Reliability
- Maintainability
- Usability
- Testability

Functional requirements

Secretary

- SEFR-01: The secretary must be able to log in
- SEFR-02: The secretary must be able to create an appointment
 - It must be mentioned if a visitor will come by car or not
 - The employee and the visitor receive an invitation for the appointment through Outlook
- SEFR-03: The secretary must be able to edit an appointment
 - The time and day of the appointment will be editable
- SEFR-04: The secretary must be able to delete/cancel an appointment
- SEFR-05: The secretary should be able to see an overview of all appointments
- SEFR-06: The secretary should be able to filter by employee name in the overview of all appointments

Admin

- AFR-01: The admin must be able to register new secretary accounts
- AFR-02: The admin must be able to edit secretary accounts' information
 - The email and password of the account will be editable
- AFR-03: The admin must be able to delete secretary accounts
- AFR-04: The admin must be able to see an overview of all secretary accounts
- AFR-05: The admin must be able to see an overview of all employees who have been added to the system
- AFR-06: The admin should be able to filter through the list of employees by name
- AFR-07: The admin must be able to upload a file containing employee names and emails to the system
 - After the file is uploaded, the employees in the file will be added to the system and will be able to have meetings scheduled for them by the secretary
- AFR-08: The admin must be able to add a singular employee's information to the system
 - The information consists of first name, last name, and email

System

- SYFR-01: System must be able to scan a license plate
 - Once a visitor's license plate has been scanned, the system sends them an SMS to tell them which parking they can park in
 - Once a visitor's license plate has been scanned, the system sends an email to the employee who has a meeting with them, telling them in approximately how much time the visitor will have arrived in the lobby, based on which parking they park in
- SYFR-02: System must be able to detect if a car is parked in a visitor parking spot in the main parking
 - Infrared sensors are used

3. Use Cases

UC number 1 – SEFR-01

UC name	User login
UC description	The user can login so it can use the functionalities of the app
Actor	Secretary
Preconditions	-Must have account that consists of an username and password
Steps	<ol style="list-style-type: none"> 1. User access the website 2. User puts the username 3. User puts the password 4. User confirms the login 5. System validates the login 6. System successfully logins the user 7. System gives a message that the user is successfully logged in
Postconditions	The user successfully logged in and redirects the user to the authorized page
Exceptions	Step 5: Fails if the user puts the wrong password/username, system throws that the user failed to login. Go back to step 2.

UC number 2 – SEFR-02

UC name	Create appointment
UC description	The user can create an appointment in the system
Actor	Secretary
Preconditions	-The user must be logged in
Steps	<ol style="list-style-type: none"> 1. System gets the data of the employee's schedule 2. User chooses the appropriate schedule for the employee 3. User fills the first name and last name of the visitor 4. User fills the E-mail 5. User fills the phone number 6. User fills if the visitor is by car 7. User confirms the appointment 8. System validates the appointment 9. System successfully added the appointment 10. System gives a message that an appointment has been successfully created
Postconditions	The appointment is successfully created in the system
Exceptions	<p>Step 8: Fails if the email is invalid, system throws to the user that the email if invalid. Go back to step 3.</p> <p>Step 8: Fails if the phone number is invalid, system throws to the user that the phone number is invalid. Go back to step 4.</p> <p>Step 8: If nothing is filled, system throws that all the blanks must be filled. Go back to step 1.</p>

Step 8: If there's already an appointment set on the specified time, system throws that there's already an appointment at the specified time. Go back to step 1.

UC number 3 - SEFR-03

<i>UC name</i>	Edit appointment
<i>UC description</i>	The user can edit an appointment in the system
<i>Actor</i>	Secretary
<i>Preconditions</i>	-The user must be logged in -The appointment must exist in the system
<i>Steps</i>	<ol style="list-style-type: none"> 1. User selects the appointment 2. User edits the appointment 3. User submits the appointment 4. System validates the appointment 5. System edits the appointment 6. System gives a message that an appointment has been successfully edited
<i>Postconditions</i>	The appointment is edited in the system
<i>Exceptions</i>	<p>Step 8: Fails if the first name/last name of the visitor is invalid, system throws to the user that the first name/last name is invalid. Go back to step 2.</p> <p>Step 8: Fails if the email is invalid, system throws to the user that the email if invalid. Go back to step 2.</p> <p>Step 8: Fails if the phone number is invalid, system throws to the user that the phone number is invalid. Go back to step 2.</p> <p>Step 8: Fails if the first name/last name of the employee is invalid, system throws to the user that the employee doesn't exists. Go back to step 2.</p> <p>Step 8: If nothing is filled, system throws that all the blanks must be filled. Go back to step 1.</p> <p>Step 8: If there's already an appointment set on the specified time, system throws that there's already an appointment at the specified time. Go back to step 1.</p>

UC number 4 - SEFR-04

<i>UC name</i>	Delete appointment
<i>UC description</i>	The user can delete an appointment in the system
<i>Actor</i>	Secretary
<i>Preconditions</i>	-The user must be logged in -The appointment must exist in the system
<i>Steps</i>	<ol style="list-style-type: none"> 1. User selects the appointment to delete 2. User confirms about the deleted appointment 3. System validates the appointment 4. System deletes the appointment 5. System gives a message that an appointment has been successfully deleted

<i>Postconditions</i>	The appointment is deleted in the system
<i>Exceptions</i>	Step 3: If the appointment doesn't exist anymore, system throws that the appointment doesn't exist. The use case ends

UC number 5 - SEFR-05

<i>UC name</i>	Overview of appointments
<i>UC description</i>	The user can see all appointments for a specified day
<i>Actor</i>	Secretary
<i>Preconditions</i>	-The user must be logged in
<i>Steps</i>	<ol style="list-style-type: none"> 1. User selects which day they want to see the appointments for 2. System pulls all the appointments of the specified day 3. System shows the scheduled appointments for the specified day
<i>Postconditions</i>	The system shows the appointments for the selected day to the user
<i>Exceptions</i>	Step 2: If there are no appointments, show to the user that there are no appointments. Go back to step 1.

UC number 5 - SEFR-05

<i>UC name</i>	Filter appointment overview
<i>UC description</i>	Filtering for appointments
<i>Actor</i>	Secretary
<i>Preconditions</i>	-The user must be logged in
<i>Steps</i>	<ol style="list-style-type: none"> 1. User fills the name 2. User confirms the filter 3. System filters the overview by name
<i>Postconditions</i>	The user can see the overview filtered by name
<i>Exceptions</i>	Step 3: If there are no appointments, show to the user that there are no appointments. Go back to step 1.

UC number 6 – AFR-01

<i>UC name</i>	Registering a new secretary
<i>UC description</i>	Adding new secretary to the system
<i>Actor</i>	Admin
<i>Preconditions</i>	- Must be logged in
<i>Steps</i>	<ol style="list-style-type: none"> 1. Admin clicks on add new secretary 2. Admin fills the username, password 3. Admin confirms the information 4. System validates the information 5. System adds the secretary
<i>Postconditions</i>	The secretary has been successfully added

<i>Exceptions</i>	Step 2: the username or password already exists, system shows that other username or password needs to be chosen. Go back to step 1.
-------------------	--

UC number 7 – AFR-02

<i>UC name</i>	Editing secretary's account information
<i>UC description</i>	Changing the information of the secretary
<i>Actor</i>	Admin
<i>Preconditions</i>	- Must be logged in
<i>Steps</i>	<ol style="list-style-type: none"> 1. User selects the secretary 2. User edits the secretary 3. User submits the secretary 4. System validates the secretary 5. System edits the secretary 6. System gives a message that an secretary has been successfully edited
<i>Postconditions</i>	The secretaries account information has been updated
<i>Exceptions</i>	

UC number 8 – AFR-03

<i>UC name</i>	Deleting a secretary account
<i>UC description</i>	Deleting an secretary account from the system
<i>Actor</i>	Admin
<i>Preconditions</i>	- Must be logged in
<i>Steps</i>	<ol style="list-style-type: none"> 1. User selects the secretary 2. User confirms the secretary about to delete 3. System deletes the secretary
<i>Postconditions</i>	The secretary has been successfully deleted
<i>Exceptions</i>	

UC number 9 – AFR-04

<i>UC name</i>	Overview secretaries
<i>UC description</i>	Overview of all secretaries
<i>Actor</i>	Admin
<i>Preconditions</i>	- Must be logged in
<i>Steps</i>	<ol style="list-style-type: none"> 1. User clicks on overview 2. System pulls all the secretaries in an overview
<i>Postconditions</i>	The overview is being shown to the user
<i>Exceptions</i>	Step 2: if there's no secretaries, system show to the user that there are no secretaries in the system. Use case ends.

UC number 10 – AFR-05

<i>UC name</i>	Overview employees
----------------	--------------------

<i>UC description</i>	Overview of all employees
<i>Actor</i>	Admin
<i>Preconditions</i>	- Must be logged in
<i>Steps</i>	3. User clicks on overview 4. System pulls all the employees in an overview
<i>Postconditions</i>	The overview is being shown to the user
<i>Exceptions</i>	Step 2: if there's no employees, system show to the user that there are no secretaries in the system. Use case ends.

UC number 11 - AFR-06

<i>UC name</i>	Filter secretary overview
<i>UC description</i>	Admin can filter the overview
<i>Actor</i>	Admin
<i>Preconditions</i>	- Must be logged in - Must have secretaries in the system
<i>Steps</i>	1. User indicates that he can filter 2. User confirms the filter 3. System filters the overview
<i>Postconditions</i>	The system successfully filtered the overview
<i>Exceptions</i>	

UC number 12 - AFR-07

<i>UC name</i>	Uploading file of employees
<i>UC description</i>	The admin can add all the information of the employee in 1 file
<i>Actor</i>	Admin
<i>Preconditions</i>	- Must be logged in - Must have a file ready
<i>Steps</i>	1. User uploads a file 2. User confirms the file 3. System checks if the file is valid 4. System adds the employees to the system
<i>Postconditions</i>	The employees are added in the system
<i>Exceptions</i>	Step 3: The file extension is invalid. System shows that the file extension is invalid. Go back to step 1. Step 3: The file has not the right format. System shows that the format inside the file is wrong. Go back to step 1.

UC number 13 - AFR-08

<i>UC name</i>	Adding an employee's information to the system
<i>UC description</i>	Adding employee to the system
<i>Actor</i>	Admin
<i>Preconditions</i>	- Must be logged in
<i>Steps</i>	1. User indicates he wants to add an employee 2. User fills the first name, last name and email

	3. User confirms the information
	4. System validates the information
	5. System adds the employee to the system
<i>Postconditions</i>	The employee has been successfully added to the system
<i>Exceptions</i>	

UC number 14 - SYFR-01

<i>UC name</i>	System scanning a license plate
<i>UC description</i>	System detecting the incoming license plate from the camera
<i>Actor</i>	System
<i>Preconditions</i>	- Connection between camera and the system
<i>Steps</i>	1. Car comes over to the field of vision 2. Camera detects the car 3. System checks the license plate
<i>Postconditions</i>	The system successfully captured the license plate of the car
<i>Exceptions</i>	

UC number 15 – SYFR-02

<i>UC name</i>	System redirecting the visitor's parking spot
<i>UC description</i>	System redirects the visitor who comes by car to the alternative parking spot
<i>Actor</i>	System
<i>Preconditions</i>	- The system knows the car is on the premise
<i>Steps</i>	1. System sends checks if there is space on the main parking spot 2. System sends a notification to the user by sms
<i>Postconditions</i>	The system successfully notified the user to where to go
<i>Exceptions</i>	Step 1: if there is space, Use case ends

4. User Stories

- **User story no.1: Log in to the platform**

Story points: 3

Priority: 10

- **As a** secretary at Sioux
 - **I can** log in to the platform
 - **So that** I can manage the appointments of the clients
-
- **Acceptance criteria:**
 - Only the authorized secretary can log in to the platform
 - Once logged in, the secretary has access to the app functionalities

- **User story no.2: Make an appointment**

Story points: 1

Priority: 98

- **As a** client of Sioux
- **I can** call the secretary
- **So that** I can make an appointment with one of the employees

- **Acceptance criteria:**
 - The client can reach the secretary by phone
 - The client makes an appointment on the spot

- **User story no.3: Make an appointment (with parking space)**

Story points: 1

Priority: 98

- **As a** client of Sioux coming by car
- **I can** call the secretary and mention that I have a car
- **So that** I can make an appointment with one of the employees and have a parking space when I have the meeting

- **Acceptance criteria:**

- When making the appointment, the client can mention that he is coming by car
- If the client comes by car, the license plate is registered in order to identify the owner

- **User story no.4: Add an appointment in the employee's schedule**

Story points: 13

Priority: 96

- **As a** secretary at Sioux
- **I can** add the appointment a client made in a employee schedule using Outlook
- **So that** both the secretary and the employee can see the schedule meetings

- **Acceptance criteria:**

- Each appointment is marked in the Outlook calendar
- The secretary can also see the scheduled appointments

- **User story no.5: View scheduled meetings in the calendar**

Story points: 5

Priority: 90

- **As an** employee at Sioux
- **I can** view all the scheduled meetings in my Outlook calendar
- **So that** I can easily know when and with whom I am meeting

- **Acceptance criteria:**

- Each employee can see only their schedule

- **User story no.6: Parking notification**

Story points: 8

Priority: 83

- **As a** client of Sioux with an appointment and coming by car
- **I can** receive a notification before my appointment
- **So that** I can know if I have a free spot in the main parking lot or I need to go in the next indicated parking

- **Acceptance criteria:**

- Each client with an appointment and coming by car receives a text message that notifies them if there are parking spots left

- If there are no spots left, the client is redirected to the next parking lot

- **User story no.7: Arrival notification**

Story points: 13

Priority: 82

- **As an** employee at Sioux
 - **I can** receive a notification when the client arrives with the car
 - **So that** I can manage my time more efficiently and be in time for the meeting
-
- **Acceptance criteria:**
 - The employees receive notifications when the clients arrive
 - If the clients come by car, the employees are notified when the car enters the parking lot

- **User story no.8: Managing meetings**

Story points: 13

Priority: 60

- **As a** secretary at Sioux
 - **I can** edit/update/ delete the meetings from the employees schedules.
 - **So that** I can manage everyone's schedule accordingly
-
- **Acceptance criteria:**
 - The employees receive notifications after a change of schedule

- The secretary can create/update/delete meetings

- **User story no.9: Filter meetings by employee**

Story points: 5

Priority: 35

- **As a** secretary at Sioux
- **I can** filter the meetings by employee
- **So that** I can see all the meetings that employee has scheduled

- **Acceptance criteria:**

- The secretary can filter the meetings by employee

- **User story no.10: Manage secretary accounts**

Story points: 13

Priority: 80

- **As a** secretary at Sioux
- **I can** filter the meetings by employee
- **So that** I can see all the meetings that employee has scheduled

- **Acceptance criteria:**

- The secretary can filter the meetings by employee

