**Software Requirements Specication (SRS)  
Project Name: shiftplannet**  
Orin Gavriel 318774338Or Goren 314619115Yehonatan Mekayten 213637424

**Introduction**

**Purpose**

ShiftPlannet is a workforce management app designed to ease scheduling and communication between employees and managers. It provides a comfortable platform where users can submit and organize shifts, request vacation/sick days, ask for shift replacements, upload important updates, and of course get immediate notifications about any move.

**Intended Audience**

* Project sponsor - an executive or from a business needing the app to improve efficiency or an investor providing financial backing for the development.
* developers- responsible for the design, development, testing, and deployment of the app.
* End-Users- the employees and managers who will use the app on a daily basis.
  + The employees will use the app to request shifts, days off and replacements, view the finalized work arrangement and receive updates from their manager.
  + The managers will use the app to arrange the shifts and publish them to the employees. They will receive requests to approve or reject with feedback and they can also publish important updates to their employees.

**Intended Use**

* **Process Name:** Submitting vacation/Sick days Requests

**Objective**: The employee wants to receive approval for vacation or sick days.

**Involved Actors**: Employee, Manager

**Prerequisites**: The employee has a user account in the app.

**Process Steps**:

The employee clicks on "vacation/sick days Request" in the app menu.

The employee adds notes and supporting documents if needed.

The employee clicks on "Submit Request".

**Successful Outcome:** The request is successfully submitted to the manager.

**Alternative Outcomes**: An error occurred and the request was not sent because the employee did not attach a sick note or did not specify the requested dates.

* **Process Name**: Submitting Shifts

**Objective**: The employee wants to submit the days and hours they are available to work in the period specified by the manager.

**Involved Actors**: Employee, Manager

**Prerequisites**: The employee has a user account in the app**.**

**Process Steps**:

The employee clicks on "Submit Constraints" in the app menu.

The employee specifies the days and hours they are available to work and adds details if necessary.

The employee clicks on "Submit Constraints"

**Successful Outcome:** The request is successfully submitted to the manager.

**Alternative Outcomes:** An error occurred and the request was not sent because the employee did not enter any details.

* **Process Name**: Creating and Publishing Work Schedule

**Objective**: The manager wants to create a work schedule and publish it.

**Involved Actors**: Manager, Employees

**Prerequisites**:

The manager has a user account in the app and permissions.

The manager has received the work constraints of all employees.

**Process Steps**:

The manager reviews the work constraints of the employees and assigns them to the schedule by days and hours.

The manager clicks on "Publish Work Schedule".

**Successful Outcome**: The work schedule is published to all employees.

**Alternative Outcomes**: An error occurred and the schedule was not sent because the manager did not assign employees to all shifts in the schedule.

* **Process Name**: Request for Shift Change  
  **Objective**: The employee wants to find someone to work in their place.  
  **Involved** **Actors**: Employees, Manager  
  **Prerequisites**: The employee has a user account in the app.  
  **Process Steps:**

The employee clicks on "Shift Change" in the app menu.

The employee specifies the day and hours they need the shift change.

The employee clicks on "Submit Shift Change Request."

The system checks who is available to work on that day but wasn't assigned a shift and sends them the shift change request.

The employees receiving the update either approve or reject the request.   
If one of the employees approves the request, the shift change request is sent to the manager for approval and change in the work management.  
If all employees reject it, the employee who submitted the request receives a notification.  
**Successful Outcome**: The request is successfully submitted to the employees and to the manager if needed.  
**Alternative Outcomes**: An error occurred, and the request was not sent because the employee did not specify the date.

* **Process Name**: Sending Updates to Employees  
  **Objective**: The manager wants to send updates via the app.  
  **Involved Actors**: Manager, Employees  
  **Prerequisites**: The manager has a user account in the app.  
  **Process Steps**:

The manager clicks on "Send Updates" in the app menu.

The manager writes the update in the system.

The manager clicks "Send Update."  
**Successful Outcome**: The update is sent to all employees.  
**Alternative Outcomes**: The system threw an error, and the update was not sent because the manager clicked "Send Update" without writing any text**.**

* **Process Name**: Approval/Denial of Employee Requests  
  **Objective**: The manager wants to send an approval/denial response to the employee's request.  
  **Involved Actors**: Manager, Employees  
  **Prerequisites**:   
  The manager has a user account in the app and permissions

The manager has received the employee's request.  
**Process Steps**:

The manager clicks on "Employee Requests" in the app menu.

The manager reviews the request and decides whether to approve or deny it.

The manager clicks on "Approve" or "Deny" and provides an explanation if necessary.

The manager clicks "Send."  
**Successful Outcome**: The decision is sent to the employee.  
**Alternative Outcomes**: The system threw an error, and the request was not sent because the manager did not select an approval/denial option for the request.

**Project scope**

Employees main features:

* Submitting vacation or sick days requests.
* submitting personal constraints for work days/hours.
* Requesting shift replacements.
* Viewing the shift arrangement after its published
* Receiving updates from the manager.

Managers main features:

* Approving or rejecting employee requests.
* Creating and publishing work schedules.
* Sending work-related updates to employees.

**Boundaries and Limitations:**

* Role Support: the system supports only two roles: Employee and Manager.
* Basic Management Capability: Managers can organize employee shifts, but the system does not support advanced features like managing overtime, payments, or payroll calculations. The app also separates itself from the company's hour count system.
* Mobile Device Support Only: The system is designed to operate on mobile android devices only and does not support web use or other mobile devices.
* Requests Cannot Be Modified Post-Submission: after submission of a request by an employee, it cannot be edited or updated.

**definitions:**

* Shift- A work period assigned to an employee during the day, such as morning, evening, or night. Shifts are not fixed daily and may change based on the work schedule and constraints.
* Constraints- Requests or limitations provided by employees regarding their working hours (for example, inability to work evenings or specific days).
* Work arrangement- A planned arrangement of employee shifts for a defined period of time (likely for each week)
* Requests- Any submission made by an employee to his manager regarding shift changes, vacation, or sick days.
* Shift Replacement- A request of an employee to switch a shift with another employee.
* Manager- A professional responsible for planning employee shifts and approving or rejecting employees' requests.

**Overall Description**

**User Needs**

The target users of the app are the employees and managers. The whole purpose is to improve the work management efficiency for them.  
The employees who use the app want it to allow them to quickly submit shifts and request shift replacements or days off. They need to get real-time updates about their request's status or any important information sent by their manager. Moreover, they need a clear view of the shift arrangement after its published.  
The managers are interested in an efficient tool to manage employee requests for vacations, sick days, shift replacements, and shift constraints and expect to receive real-time notifications about the submitted requests. Moreover, managers should be able to publish the finalized work arrangement in the app and important updates.  
The app should offer both target users a work management tool with clarity and transparency.  
It should save time in manual scheduling or back-and-forth communication through several platforms and minimize errors such as human errors, gaps in shift arrangement and delays.  
In addition, the app, which is used in mobile phones, provides its users the most convenient way to access schedules, submit requests, and communicate in real-time in a user-friendly platform and also combines all work management processes into one app to improve efficiency.

**Assumptions and Dependencies**

* The app is designed to work exclusively on Android mobile phones
* The app is displayed to users in Hebrew
* every employee must have an account in the system
* The app will require internet services for updates, real time data, notifications, login etc.
* The app requires certain device permissions as access to notifications and network connectivity
* The app is developed using Android Studio in java language
* The app uses Firebase for its database and other services as real-time data synchronization, cloud massaging and user authentication.

**System Features and Requirements**

**Functional Requirements and System Features**

* **Secure registration and login:**
  + The app provides different access levels, ensuring employees and managers have tailored tools and permissions to meet their specific needs.
  + New employees can create accounts using username and passward with optional integration for Google or other third-party authentication providers.
  + Existing users can log in easily likewise.
* **Vacation and Sick days Requests:**
  + Employees can easily request vacation or sick days through the app.
  + Employees can add a photo of the doctor's note in case of a sick day.
  + Managers receive instant notification, enabling them to approve or decline requests while providing feedback.
  + Employees are notified of their manager’s decision.
* **Shift Management:**
  + Employees can submit their availability for shifts through the app.
  + Managers can organize, finalize, and publish shift schedules, with employees receiving notifications upon publication.
  + Shift replacement requests allow employees to seek replacements, notifying colleagues who can accept the request. Managers oversee all changes, with the ability to approve or deny them.
* **Work-related Updates:**
  + Managers can share important work-related announcements or updates.
  + Employees receive real-time notifications for every update published.
* **Push Notifications:**Real-time notifications keep employees and managers informed about shift changes, requests, approvals, and updates.

**External Interface Requirements**

* **User Interface:**The app will provide its visitors a user-friendly platform for shift management supported by android mobile phones. The interface will be simple and easy to manage for all users withClear icons and labels and of course push notifications for real-time updates.
* **Third-party Integrations and Cloud Storage:**The app requires integration with Firebase services for:
* Authentication- To manage secure employee logins
* Firebase real-time db-  For instant updates on shift changes and notifications.
* Firebase Cloud Messaging- To send real-time updates and reminders to employees.
* Firebase Cloud Firestore- to store and synchronize data such as employee details, shift schedules files storage .

**Nonfunctional Requirements**

* **Performance:**  
  The app must work successfully on all android mobile phones.  
  The system should load quickly and respond to user actions within seconds without delays. Data processing tasks should be optimized to ensure best performance, even during peak usage.  
  Moreover, the app should be user-friendly with clear and quick navigation and use.
* **Scalability:**The system should be able to handle an increased number of users, features, data and traffic, especially at peak times, without compromising performance and of course do it automatically without manual intervention.
* **Security:**The app must comply with data protection regulations, ensuring secure user data storage and communication.  
  Firebase Authentication will be used for secure user registration and sign-ins. Firebase's security rules will control data access, ensuring only authorized users can access relevant information based on their roles
* **Availability:**.  
  The system should ensure immediate availability at all times.  
   It will use firebase services designed for reliability and automatically scaling to handle increased traffic and minimize downtime.

**Specific Needs and requirements of End Users**

**Employees**

* Characteristics:
* Frontline workers responsible for executing tasks during assigned shifts.
* May work part-time or full-time, with varied availability.
* Use:
* Shift Management:

View personal shift schedules and team-wide calendars.

Submit availability for future schedules.

* Request submission:  
  Request vacation or sick days directly through the app.  
  Initiate a replacement request if unable to attend an assigned shift.
* Communication:  
  Receive notifications about approvals, rejections, or updates on shifts and requests.
* Access:
* Limited access to their own schedules, shift details, and status of submitted requests.
* Can view team-wide calendars
* Customized Views:  
  Personalized dashboard displaying upcoming shifts, pending requests and manager updates.
* Specific Functionalities:
* Ability to request vacation or sick days with optional notes/documents attached for manager review.
* Option to initiate shift replacement requests by notifying manager or other employees through the app.
* Notifications for request approvals/denials, schedule changes, or replacement confirmations.

**Managers**

* Characteristics:
* Oversee and coordinate shifts for all employees.
* Ensure sufficient shift coverage and maintain operational efficiency.
* Use:
* Schedule Creation and Oversight:  
  Assign shifts to employees based on availability and requirements.
* Request Management:  
  Review and approve/deny vacation and sick day requests.  
  Handle shift replacement requests, either by assigning replacements or approving suggested employees.
  + publishing updates:  
    publish day to day updates to employees  
    publish the finalized work arrangement
* Access:
  + Full access to all employee schedules, including details like availability, requests, and shift coverage.
  + Administrative privileges to modify schedules and approve or deny requests.
* Customized Views:
  + Managerial dashboard featuring:
  + Pending vacation/sick day requests
  + Uncovered shifts flagged for attention.
* Specific Functionalities:
  + Tools to manage and approve replacement requests, with auto-suggestions based on employee availability.
  + scheduling capabilities for assigning shifts efficiently.
  + Ability to communicate updates to employees via in-app messaging or notifications.

**Change Management**

To ensure clarity and accountability, changes to the document will be managed as follows:

* **Version Control System:**Every change made in the document will result in a new version .  
  The document will be maintained in a version control system (for example Git or shared collaborative tools like Google Docs) to track and store changes.  
  each change .  
  A structured version history table (as attached) will document every modification, including the version number, date of change, name of the responsible party and a short description of the changes made.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Responsible Party** | **Description of Change** |
| **1.0** | **YYYY-MM-DD** | **Initial Author** | **Initial draft created.** |
| **1.1** | **YYYY-MM-DD** | **Contributor Name** | **Minor edits** |
| **2.0** | **YYYY-MM-DD** | **Document Owner** | **Major updates** |

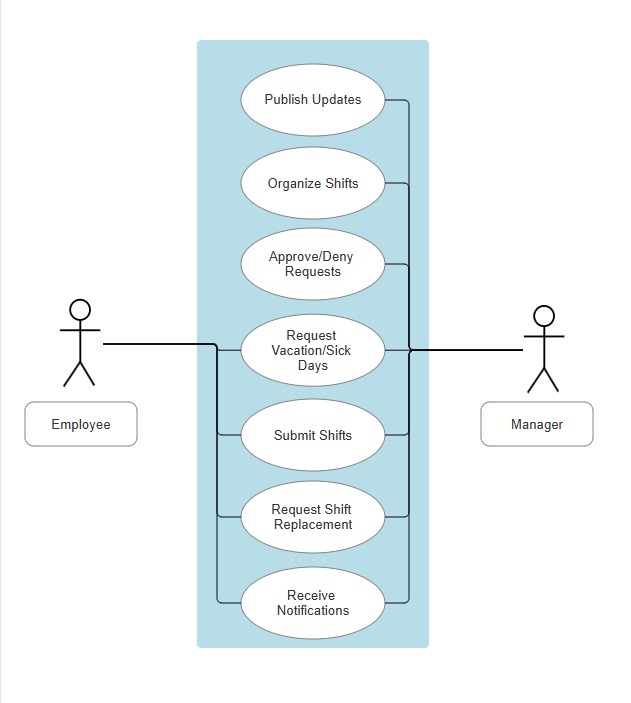
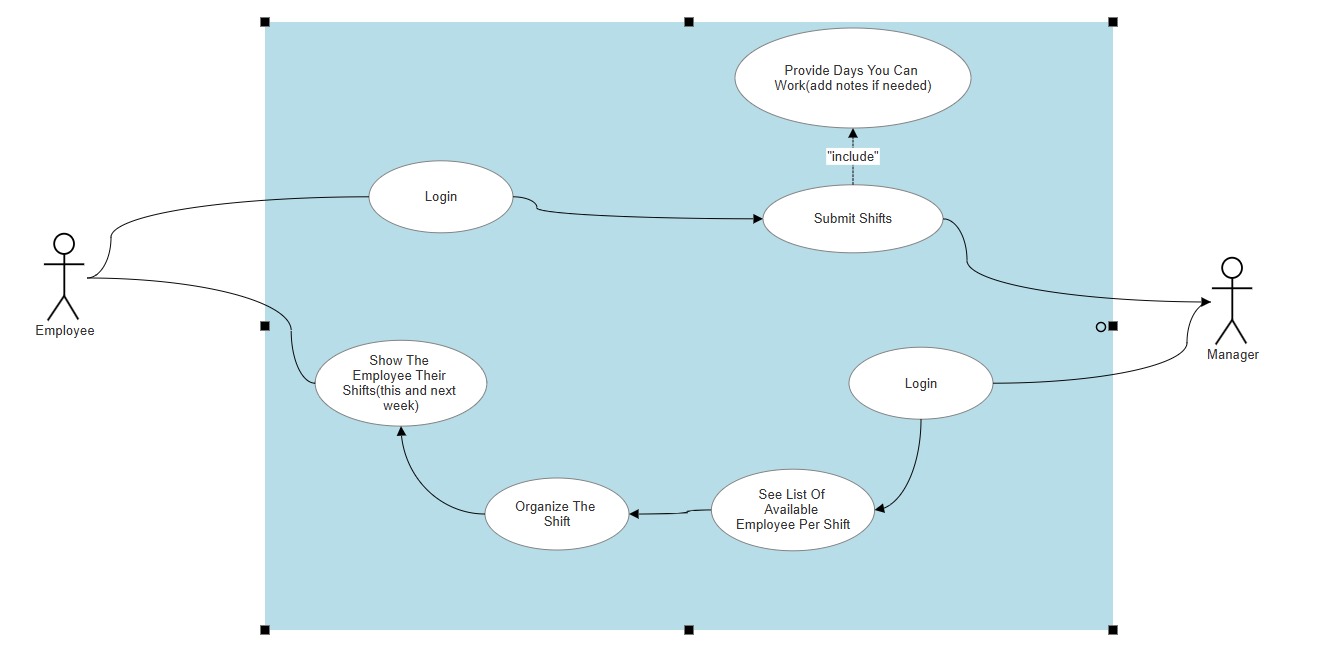
**the table document will be used as follows:**

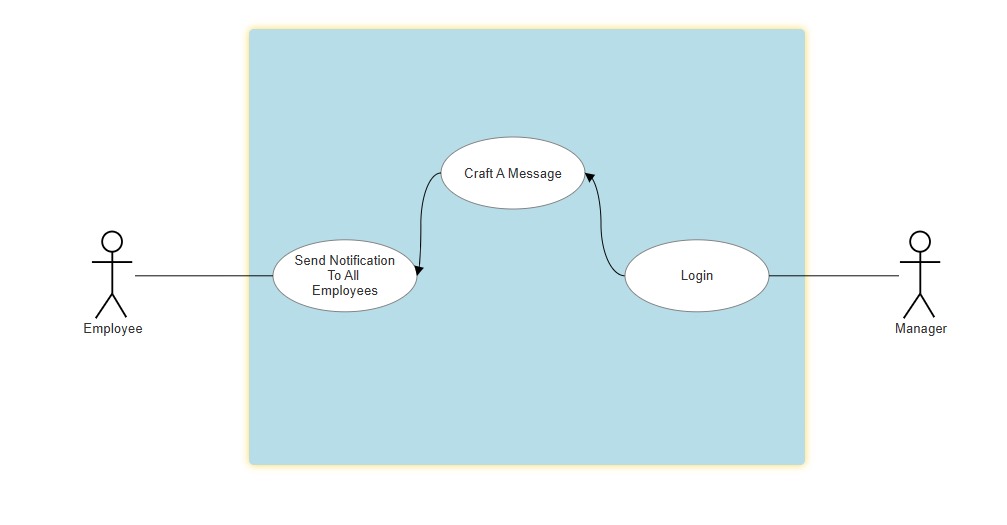
1. Incremental version numbers for minor edits (e.g., 1.0 → 1.1).
2. Whole number increments for major changes (e.g., 1.1 → 2.0).
3. Clear and concise descriptions to ensure quick reference for future reviews.

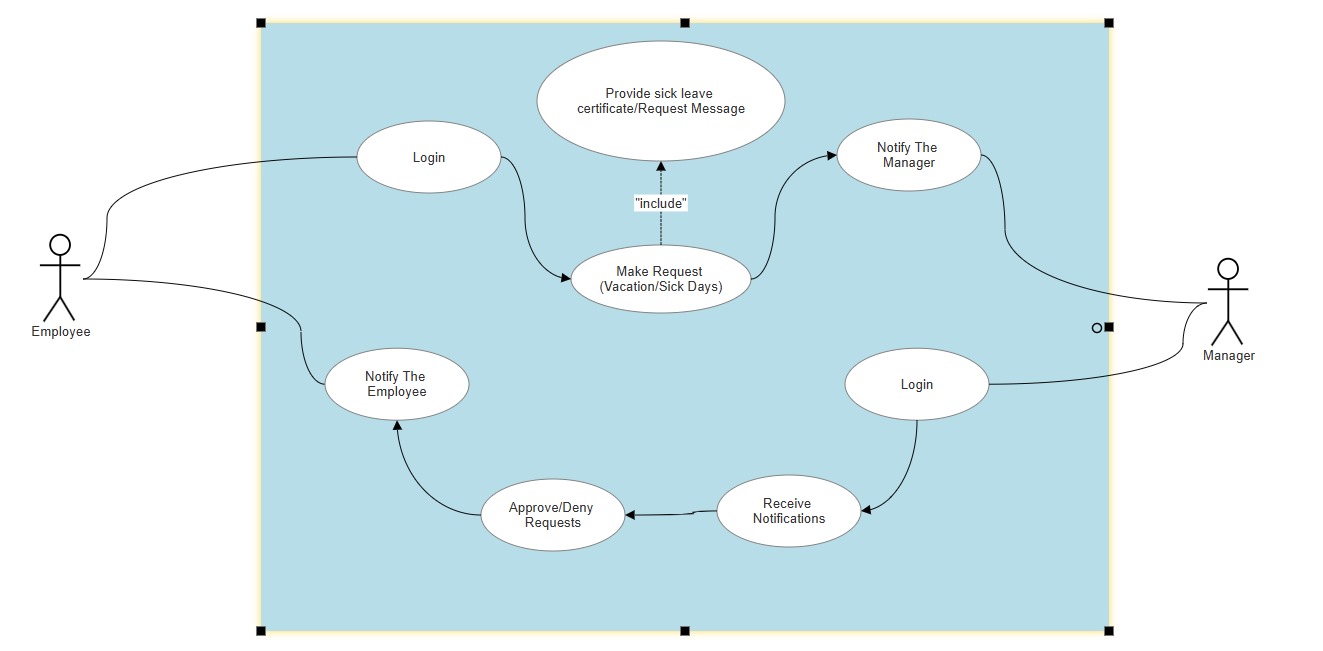
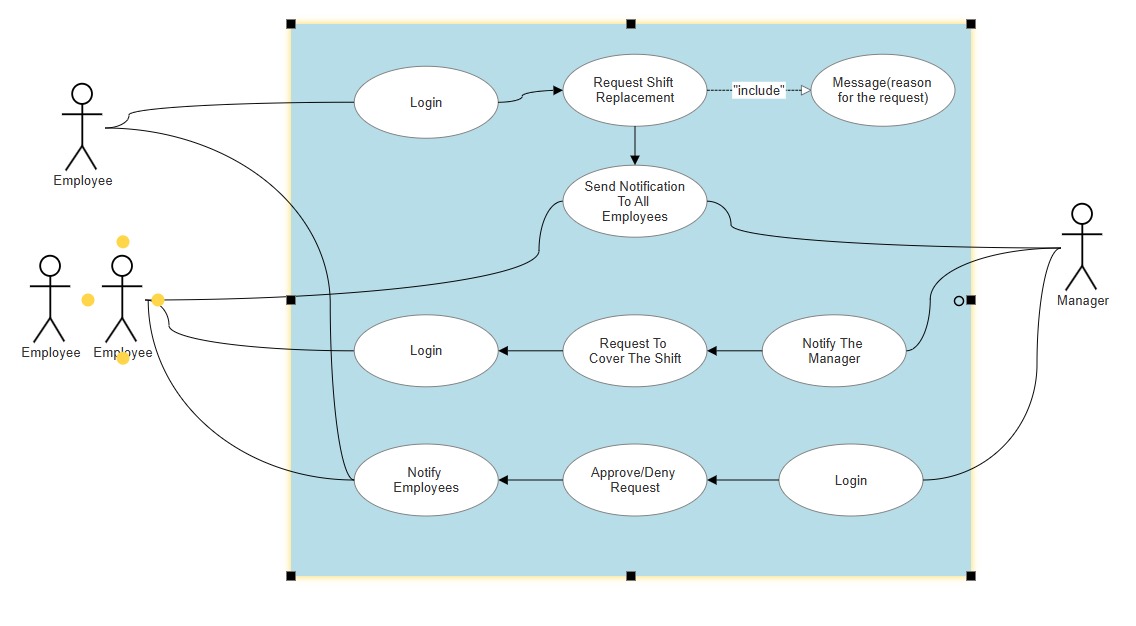
By using this version control table, the document will remain reliable and transparent throughout the app’s lifecycle.

* **Approval Process and responsible parties**:  
  Before making any changes to the document or system, approval must be obtained from a manager or responsible party authorized.
* **Notification to users:**Changes will be communicated to all relevant users in the app’s notification system.

**Appendices- use case diagrams**

* ****general:
* submitting shifts + creating and publishing work schedule:
* Sending updates to employees:



* ****Submitting vacation/sick days request:
* Shift replacement: