### בס"ד

### Part B - Quality Strategy:

1. Requirements Review (RR):

**Functional Requirements:**

1. **Creating Group Isolation:**
   * Ensure clear instructions for initiating group isolation, including the steps involved and expected outcomes.
   * Specify how the group members are identified and updated during the isolation period.
   * Consider the scenario of a person who was not infected but was in the radius of the patient, potentially vaccinated or previously sick, and clarify how such cases are handled within the group isolation process.
2. **Creating a Google Maps Component:**
   * Clarify the integration process with Google Maps API, including obtaining and displaying geographic data accurately.
   * Specify how the map component interacts with other system functionalities, such as selecting locations for group isolation.
   * Address the scenario of entering an incorrect address, where users input a different address than the real address, and detail the system's response to such discrepancies.
3. **Creating a Calendar for Date of Exposure to the Virus:**
   * Detail the implementation of the calendar feature, including user interaction and data storage.
   * Ensure proper validation for date inputs to prevent errors and inconsistencies.
4. **Updating Isolated People as Required:**
   * Define the process for updating the array of isolated people, including criteria for inclusion and removal.
   * Specify how the system handles updates during the isolation period and upon recovery.
   * Account for the scenario where a person changes their phone number, which may not be updated in the phone numbers of the entire population of Israel, potentially resulting in messages being sent to the wrong person.

**Disadvantages/Questions:**

* **Missing Calculation for Recovery Date in the UI:**
  + Clarify how the recovery date is calculated and displayed to users within the UI.
  + Ensure consistency and accuracy in presenting recovery information to users.
* **Finding Isolated People:**
  + Provide guidance on how the system identifies and retrieves information about isolated individuals.
  + Specify the criteria for selecting and updating isolated people based on exposure events.
* **Updating Isolated People with Isolation Status:**
  + Define the process for updating the isolation status of affected individuals within the system.
  + Ensure proper communication channels for notifying isolated individuals and monitoring their status.
* **Communication Channels for Notifications:**
  + Specify the channels (e.g., phone, email, SMS) used for notifying isolated individuals and relevant authorities.
  + Ensure compliance with data privacy regulations and user preferences for communication.
* **Status of Quarantined Persons:**
  + Clarify how the system tracks and displays the status of quarantined individuals, including their compliance with isolation protocols.
* **Permissions Segmentation:**
  + Define roles and permissions within the system for administrators, system administrators, and employees.
  + Specify access levels and capabilities for each user role to maintain security and accountability.
* **Personal Details Retention for Investigations:**
  + Address the retention and protection of personal details collected for investigative purposes, ensuring compliance with data privacy laws.

**Non-functional Requirements:**

* Ensure the system is user-friendly, secure, scalable, reliable, and compatible with various devices and platforms.

2. Feature Testing Document:

**Server Side:**

* Validate inputs for accuracy and completeness, including proper ID verification and data type formatting.
* Confirm data storage and retrieval processes, ensuring integrity and consistency.
* Verify the accuracy of recovery date calculations and the reasonableness of isolation requirements based on geographic data.
* Ensure availability of contact methods for each location requiring isolation.

**Client Side:**

* Test interface compatibility across different browsers and operating systems to ensure consistent user experience.
* Verify functionality of date feature, including scrolling and selection within the calendar component.
* Assess the effectiveness of the permission system in controlling user access and actions within the UI.

**Position (End User Device):**

* Check compatibility between application server and end user devices, ensuring proper activation and functionality.
* Test resolution adaptation to accommodate various screen sizes and resolutions on end user devices.
* Verify connectivity between application server, database server, and end user devices, including open port checks for seamless communication.

**Additional Testing Considerations:**

* Performance Testing: Assess system performance under various loads to ensure responsiveness and scalability.
* Security Testing: Conduct vulnerability assessments and penetration testing to identify and address potential security risks.
* Compliance Testing: Validate adherence to relevant regulations and standards.
* Usability Testing: Gather feedback from users to evaluate ease of use, intuitiveness, and overall user satisfaction with the feature.