Contents

[Project Requirements 2](#_Toc137544361)

[Functional Requirement 3](#_Toc137544362)

[Non Functional Requirement 9](#_Toc137544363)

[Use Case Scenarios 10](#_Toc137544364)

[FR01 New User Registration 10](#_Toc137544365)

[FR02 Login 11](#_Toc137544366)

[FR03 User Email Update 13](#_Toc137544367)

[FR04 Periodic Password Reset 15](#_Toc137544368)

[FR05 Incorrect Password Lock 15](#_Toc137544369)

[FR0X BlaBla 16](#_Toc137544370)

# Project Requirements

|  |  |
| --- | --- |
| Req. ID | Description |
| PR01 | The ability for the user to register with the system. This specific requirement has the following subcomponents. |
| PR01A | The user shall provide the system their email address and password. The system shall ensure that the user’s password is secure. Passwords must be a minimum of 10 characters long and include one number and one symbol. |
| PR01B | The system shall use two-factor authentication. |
| PR01C | The system should force a user to periodically reset their password (at least once a month). |
| PR01D | If the user enters the incorrect password more than three times, then their account will be locked for one hour. |
| PR01E | The system shall allow users to change their email address in a secure way. |
| PR01F | The system shall provide the ability for the user to recover or change their password in the event it is lost. |
| PR02 | After successful registration the system shall assign the user a unique contact identification name (contact identifier) ; this can be the user’s  email address, or some other name chosen by the user if it does not conflict with other user’s contact identifiers already in the system. |
| PR03 | The system shall provide a contact list that associates a person with their contact identifier (last name, first name, address, e-mail, contact identifier). When a contact is associated with a contact identifier the VoIP application shall display the contact’s name instead of the contact identifier. |
| PR04 | The system shall provide the ability to initiate a call using a contact identifier or the contacts list. During the call initiation, the user shall be presented with call status and outcome (answered, busy or rejected).  During call initiation the user shall have the ability to end the call at any time. |
| PR05 | The system shall provide the ability to accept or reject calls while not in a call. Application shall show the caller’s contact identifier or contact name during an incoming call. |
| PR06 | The system shall notify the user of missed calls, either because the call was not accepted or because the called entity was in another call. |
| PR07 | Provide the ability to terminate a call at any time while in a call. If a call is terminated by one user, the other caller shall be notified. |
| PR08 | Application shall be brought to the foreground during an incoming call. |
| PR09 | This application is a point-to-point communication system. That is, each end point of the call should function as both a server and a client. |
| PR10 | Performance: The system must deliver call video/audio as close to real time as possible. |
| PR11 | Authentication: The system must use two factor authentication for sign on and user credentials must be protected. Lost or compromised credentials must be handled in a reasonable way. |
| PR12 | Communication privacy: The system must ensure that calls remain private.  No intermediary should be able to snoop or spy on an ongoing call. |
| PR13 | Proof of identity (nonrepudiation): Users should be confident that the entity they are on a call with is the one that they believe it is. |
| PR14 | Reliability: The system must ensure that calls are reliable. The system should recover from networking errors and dropped calls as soon as possible. The goal is to maintain a secure, performant connection at all costs. |

# Functional Requirement

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req. ID | Req. Name | Sub Req. ID | Description | PR ID | Test ID | Implemented by |
| FR01 | new user registration | FR01REG01 | The system shall provide new user registration form that captures essential information such as first name, last name, email address, OTP and password. | PR01B | TC001 | App |
|  |  | FR01REG02 | The system shall implement validation to ensure the uniqueness of email addresses to prevent multiple users from registering with the same credentials. | PR01 | TC002, TC003 | Backend |
|  |  | FR01REG03 | The system shall send a verification email to the user's provided email address upon registration to confirm their ownership and prevent misuse. | PR01F | TC002, TC003 | Backend |
|  |  | FR01REG04 | The system shall include an OTP in the email to be sent to verify their account. | PR01B | TC007 | Backend |
|  |  | FR01REG05 | The system shall implement a mechanism to handle expired or revoked verification OTP |  | TC006 | Backend |
|  |  | FR01REG06 | Passwords must be a minimum of 10 characters long and include one number and one special character. | PR01A | TC004 | Backend |
|  |  | FR01REG07 | Passwords must be confirmed through re-type. |  | TC005 | Backend + App |
|  |  | FR01REG08 | The system shall hash and salt the passwords before storing them in the database to enhance security. |  |  | Backend |
|  |  | FR01REG09 | The system shall provide meaningful error messages and validation feedback to users during the registration process to assist them in resolving any issues they encounter. |  | TC008 | Backend |
|  |  | FR01REG10 | The system shall implement error logging and monitoring to track and investigate any registration-related errors or anomalies. |  | TC008 | Backend |
|  |  | FR01REG11 | The system shall implement CAPTCHA or other anti-bot mechanisms to prevent automated registrations. ( Optional ) |  | TC009 |  |
| FR02 | Login | FR02LOG01 | The user shall provide the system their email address, password and OTP. | PR01A, PR01B | TC010 | App |
|  |  | FR02LOG02 | The email address should be in a standard format. (e.g., "[example@example.com](mailto:example@example.com)"). | PR01A | TC011 | Backend + App |
|  |  | FR02LOG03 | The password information must be masked and show | PR01A |  | App |
|  |  | FR02LOG04 | The system should provide appropriate error messages or notifications to the user if there are any issues with the provided email address or password. |  | TC012 | Backend + App |
|  |  | FR02LOG05 | If email and password are corrected, the user can request OTP | PR01B | TC013 | Backend + App |
|  |  | FR02LOG06 | The OTP will be sent to the user's email. | PR01B | TC014,TC015 | Backend |
|  |  | FR02LOG07 | The OTP must be 6 characters long and consist of numbers only. | PR01B | TC014,TC015 | Backend |
|  |  | FR02LOG08 | The system should provide appropriate error messages or notifications to the user if there are any issues with the OTP. | PR01B | TC016 | Backend + App |
| FR03 | user email update | FR03UPD01 | Only authorized users should be able to access the functionality to change their email address. | PR01E | TC017 | Backend |
|  |  | FR03UPD02 | Before allowing users to change their email address, the system should authenticate their identity to ensure they are the legitimate account holders using password. | PR01E | TC018 | Backend |
|  |  | FR03UPD03 | The system should validate the user-provided email address to ensure it follows the correct format. | PR01E | TC019 | Backend + App |
|  |  | FR03UPD04 | The system should send a verification OTP to the new email address provided by the user. | PR01E | TC020 | Backend |
|  |  | FR03UPD05 | The system should require the user to confirm the change by entering the OTP within a specified timeframe | PR01E | TC020 | Backend |
|  |  | FR03UPD06 | The system should implement proper error handling mechanisms to handle scenarios such as invalid email addresses, database failures, network errors, or other exceptional cases. | PR01E | TC021 | Backend |
|  |  | FR03UPD07 | Ensure the audit trail is securely stored and accessible only to authorized personnel for monitoring, auditing, and investigating purposes. | PR01E | TC021 | Backend |
|  |  | FR03UPD08 | The system should notify users via their existing email address when a change to their email address is requested or successfully completed. | PR01E | TC017 | Backend |
| FR04 | Periodic PW Reset | FR04PWRST01 | The system shall store the date and time of the user's last password reset | PR01C |  | Backend |
|  |  | FR04PWRST02 | The system shall enforce a maximum password lifetime of one month (30 days) | PR01C | TC022 | Backend |
|  |  | FR04PWRST03 | The system shall compare the current date with the user's last password reset date to determine if a password reset is required | PR01C | TC022 | Backend |
|  |  | FR04PWRST04 | If the time since the last password reset exceeds one month, the system shall prompt the user to reset their password | PR01C | TC023 | Backend |
|  |  | FR04PWRST05 | The system shall display a notification to the user when their password is due for a reset | PR01C | TC023,TC024 | App |
|  |  | FR04PWRST06 | The system shall validate and confirm the new password entered by the user to ensure accuracy | PR01C | TC025 | Backend + App |
|  |  | FR04PWRST07 | The system shall log the date and time of the password reset for auditing and security purposes | PR01C | TC027 | Backend |
|  |  | FR04PWRST08 | The system shall send a notification email to the user after a successful password reset, confirming the password change | PR01C | TC026 | Backend |
|  |  | FR04PWRST09 | The system shall store the history of the user's previous passwords to prevent reuse of the same or similar passwords within a specified period (e.g., the last five passwords) | PR01C | TC028,TC029 | Backend |
|  |  | FR04PWRST10 | The system shall provide an option for users to contact support if they encounter any issues during the password reset process or have concerns about their password security | PR01C | TC027 | Backend + App |
| FR05 | Lockout due to an incorrect password | FR05LOCK01 | The system shall track the number of failed login attempts for each user. | PR01D | TC031 | Backend |
|  |  | FR05LOCK02 | The system shall increment the failed login attempt count by one each time a user enters an incorrect password. | PR01D | TC030 | Backend |
|  |  | FR05LOCK03 | The system shall reset the failed login attempt count to zero if the user successfully logs in. | PR01D | TC031 | Backend |
|  |  | FR05LOCK04 | The system shall lock a user's account if the failed login attempt count exceeds a predefined threshold (e.g., three). | PR01D | TC032 | Backend |
|  |  | FR05LOCK05 | The system shall enforce a lockout duration of one hour for a locked account | PR01D | TC033 | Backend |
|  |  | FR05LOCK06 | The system shall display an appropriate error message to the user when their account is locked due to excessive failed login attempts. | PR01D | TC035 | Backend + App |
|  |  | FR05LOCK07 | The system shall prevent a locked account from being accessed during the lockout duration, regardless of the password entered | PR01D | TC033 | Backend |
|  |  | FR05LOCK08 | The system shall display a countdown timer indicating the remaining lockout duration for the user | PR01D | TC033 | App |
|  |  | FR05LOCK09 | The system shall automatically unlock the user's account after the lockout duration has elapsed | PR01D | TC034 | Backend |
|  |  | FR05LOCK10 | The system shall notify the user via email when their account is locked due to excessive failed login attempts | PR01D | TC035 | Backend |
|  |  | FR05LOCK11 | The system shall include a link in the email notification for the user to contact support if they believe their account has been locked incorrectly or for any other account-related issues. | PR01D | TC035 | Backend |
|  |  | FR05LOCK12 | The system shall log all account lockout events for auditing and security purposes | PR01D | TC032 | Backend |
|  |  | FR05LOCK13 | The system shall provide an option for users to reset their password during the account lockout period using the password recovery functionality. | PR01D | TC036 | Backend + App |
| ~~FR06~~ | ~~Reset PW~~ | ~~FR06PWRST01~~ | ~~The system shall provide a form for users to initiate password recovery.~~ | PR01F | TC037,TC039 | App |
|  |  | ~~FR06PWRST02~~ | ~~The system shall show the user to enter their registered email address.~~ | PR01F | TC037 | App |
|  |  | ~~FR06PWRST03~~ | ~~The system shall validate the entered email address and verify its existence in the user database.~~ | PR01F | TC037,TC038 | Backend |
|  |  | ~~FR06PWRST04~~ | ~~If the email address is valid and registered, the system shall generate a temporary password and send it to the user's email address.~~ | PR01F | TC037,TC038,  TC040 | Backend |
|  |  | ~~FR06PWRST05~~ | ~~The system shall provide a secure password reset form where the user can enter a new password after the user successfully login using the temporary password~~ | PR01F | TC039 | Backend + App |
|  |  | ~~FR06PWRST06~~ | ~~Upon successful password reset, the system shall notify the user via email that their password has been changed.~~ | PR01F | TC039 | Backend |
|  |  | ~~FR06PWRST07~~ | ~~The system shall log all password recovery and reset activities for auditing and security purposes.~~ | PR01F | TC041~TC043 | Backend |
| FR07 | Unique ID | FR07UID01 | After successful registration the system shall assign the user a unique contact identifier. | PR02 | TC044 | Backend |
| FR08 | Contact list | FR08CTT01 | The system shall provide a contact list that associates a person with their contact identifier (last name, first name, e-mail, contact identifier). | PR03 | TC044 | Backend |
|  |  | FR08CTT02 | When a contact is associated with a contact identifier the VoIP application shall display the contact’s name instead of the contact identifier. | PR03 | TC045 | App |
| FR09 | call | FR09CALL01 | The system shall provide the ability to initiate a call using a contact identifier or the contacts list. | PR04 | TC046 | App |
|  |  | FR09CALL02 | The system shall maintain a log of call activities, including call start time, duration, participants, and call outcome (answered, busy, or rejected). | PR04 | TC046 | Backend |
|  |  | FR09CALL03 | Provide a call history feature that allows users to view and review past calls, including details like participants, timestamps, and call duration. | PR04 | TC047 | App |
|  |  | FR09CALL04 | During the call initiation, the user shall be presented with call status and outcome (answered, busy or rejected). | PR04 | TC048 | App |
| FR10 | connection | FR10CON01 | The system shall provide the ability to accept or reject calls while not in a call. | PR05 | TC050,TC051 | App |
|  |  | FR10CON02 | Application shall show the caller’s contact identifier or contact name during an incoming call. | PR05 | TC050 | App |
| FR11 | notice | FR11NOTI01 | The system shall notify the user of missed calls, in case of the call was not accepted. | PR06 | TC051,TC0525 | App |
|  |  | FR11NOTI02 | The system shall notify the user of missed calls, when the called entity was in another call. | PR06 | TC053 | App |
| FR12 | disconnect | FR12DISC01 | Provide the ability to terminate a call at any time while in a call. | PR04,PR07 | TC049 | App |
|  |  | FR12DISC01 | If a call is terminated by one user, the other caller shall be notified. | PR07 | TC054 | App |
| FR13 | Activation | FR13ACT01 | Application shall be brought to the foreground during an incoming call. | PR08 | TC055 | App |
| FR14 | Communication methods | FR14CMM01 | This application is a point-to-point communication system. That is, each end point of the call should function as both a server and a client. | PR09 | TC056 | App |

# Non Functional Requirement

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req. ID | Req. Name | Func ID | Func Name | Description | PR ID |  |
| NFR01 | Performance | NFR01\_PERF01 | Real time communication | The system must deliver call video/audio as close to real time as possible. | PR10 |  |
| NFR02 | Authentication | NFR02\_AUTH01 |  | The system must use two factor authentication for sign on and user credentials must be protected. | PR11 |  |
|  |  | NFR02\_AUTH02 |  | Lost or compromised credentials must be handled in a reasonable way. | PR11 |  |
| NFR03 | communication privacy | NFR03\_PRI01 | Privating communication | The system must ensure that calls remain private. No intermediary should be able to snoop or spy on an ongoing call. | PR12 |  |
| NFR04 | nonrepudiation | NFR04\_NREP01 |  | Users should be confident that the entity they are on a call with is the one that they believe it is. | PR13 |  |
| NFR05 | reliability | NFR05\_REL01 |  | The system must ensure that calls are reliable. The system should recover from networking errors and dropped calls as soon as possible. The goal is to maintain a secure, performant connection at all costs. | PR14 |  |

# Use Case Scenarios

## FR01 New User Registration

|  |  |
| --- | --- |
| Req ID | FR01 |
| Title | New User Registration |
| Primary Actor | User |
| Pre-conditions | * The user must have access to the registration form provided by the system. * The system must be operational and able to handle registration requests. * The user must have a valid email address to receive the verification email. * The user must have a secure internet connection to access the registration form and receive the verification email. * The database server must be accessible and able to store user data. |
| Scenario | 1. User opens the registration form of the system. 2. System presents the registration form to the user, including fields for first name, last name, email address, OTP, and password. 3. User enters their registration information. 4. System validates the uniqueness of the email address. 5. If the email address is already registered, the system displays an error message to the user. 6. If the email address is unique, the system generates a unique identifier for the user. 7. System sends a verification email to the user's provided email address, including an OTP. 8. User checks their email and retrieves the OTP. 9. User enters the OTP into the registration form. 10. System validates the OTP and checks for expiration or revocation. 11. If the OTP is invalid, expired, or revoked, the system displays an error message to the user. 12. If the OTP is valid, the system enforces password requirements. 13. User enters and confirms their password, ensuring it meets the requirements. 14. System hashes and salts the password for secure storage. 15. System stores the user's registration data in the database. 16. System displays a success message to the user, confirming their registration. 17. User can now log in to the system using their registered email address and password. |
| Post-conditions | * Upon successful registration, the user's information is stored securely in the database. * The user receives a verification email containing an OTP to verify their account. * If the OTP is validated successfully, the user is considered registered and can proceed with logging into the system. * The user can now access the system using their registered email address and password. |
| Alternate Flow | 1. User opens the registration form of the system. 2. System presents the registration form to the user, including fields for first name, last name, email address, OTP, and password. 3. User enters their registration information. 4. System validates the uniqueness of the email address. 5. If the email address is already registered, the system displays an error message to the user. 6. If the email address is unique, the system encounters an error while generating a unique identifier. 7. System displays an error message to the user, indicating the problem with generating a unique identifier. 8. User can choose to retry the registration process or contact support for assistance. 9. If the user chooses to retry, the process continues from Step 1. 10. If the user chooses to contact support, the process is temporarily halted until the issue is resolved. 11. Once the issue is resolved, the user can proceed with the registration process from the beginning. |

## FR02 Login

|  |  |
| --- | --- |
| Req ID | FR02 |
| Title | Login |
| Primary Actor | User |
| Pre-conditions | * The user must have a registered account with the system. * The user must have a valid email address and password. * The user must have access to their registered email account. * The system must be operational and accessible. |
| Scenario | 1. User opens the login page of the system. 2. System presents the login form to the user, including fields for email address, password, and OTP. 3. User enters their email address and password. 4. System validates the email address and password format. 5. If the email address or password is invalid, the system displays an error message to the user. 6. If the email address and password are valid, the user can request an OTP. 7. User clicks the "Request OTP" button. 8. System generates an OTP and sends it to the user's registered email address. 9. User checks their email and retrieves the OTP. 10. User enters the OTP into the login form. 11. System validates the OTP format and checks for expiration. 12. If the OTP is invalid or expired, the system displays an error message to the user. 13. If the OTP is valid, the system authenticates the user. 14. System logs the user into the system and grants access to the authorized features. 15. System displays a success message to the user, confirming their login. 16. User can now interact with the system and perform the desired actions. |
| Post-conditions | * The user is successfully logged into the system. * The user has access to the authorized features and functionalities. * The system tracks the user's session and activity. * The user can perform actions within their authorized scope. * If the user encounters any issues during the login process, appropriate error messages are displayed, and the user is guided to resolve the issues. |
| Alternate Flow | 1. User opens the login page of the system. 2. System presents the login form to the user, including fields for email address, password, and OTP. 3. User enters their email address and password. 4. System validates the email address and password format. 5. If the email address or password is invalid, the system displays an error message to the user. 6. If the email address and password are valid, the user can request an OTP. 7. User clicks the "Request OTP" button. 8. System encounters an issue sending the OTP to the user's email address. 9. System displays an error message to the user, indicating the problem with the email delivery. 10. User can choose to retry the OTP request or contact support for assistance. 11. If the user chooses to retry, the process continues from Step 7. 12. If the user chooses to contact support, the process is temporarily halted until the issue is resolved. 13. Once the issue is resolved, the user can proceed with the login process from the beginning. |

## FR03 User Email Update

|  |  |
| --- | --- |
| Req ID | FR03 |
| Title | User Email Update |
| Primary Actor | User |
| Pre-conditions | * The user must be an authorized user with access to the email update functionality. * The user must have a valid account and be logged into the system. * The user must have an existing email address associated with their account. * The user must have a secure internet connection to access the email update functionality. * The system must be operational and able to handle email update requests. |
| Scenario | 1. User initiates the email update process. 2. System authenticates the user's identity by requesting their password. 3. User enters their password. 4. System validates the password and confirms the user's identity. 5. System presents the email update form to the user. 6. User enters the new email address they wish to update to. 7. System validates the new email address format. 8. If the new email address is invalid, the system displays an error message to the user. 9. If the new email address is valid, the system sends a verification OTP to the new email address. 10. System displays a success message indicating that the verification OTP has been sent. 11. User checks their new email address and retrieves the OTP. 12. User enters the OTP within the specified timeframe. 13. System verifies the OTP and confirms the email address change. 14. If the OTP is incorrect or expired, the system displays an error message to the user. 15. If the OTP is correct and within the specified timeframe, the system updates the user's email address. 16. System securely stores the audit trail of the email address change. 17. System sends a notification email to the user's existing email address, informing them of the email address change. 18. System displays a success message to the user, confirming the email address change. |
| Post-conditions | * The user's email address is successfully updated in the system. * The user receives a notification email informing them of the email address change. * The user can now use the new email address for authentication and communication. * If the user encounters any issues during the email update process, appropriate error messages are displayed, and the user is guided to resolve them. |
| Alternate Flow | 1. User initiates the email update process. 2. System authenticates the user's identity by requesting their password. 3. User enters their password. 4. System validates the password and confirms the user's identity. 5. System presents the email update form to the user. 6. User enters the new email address they wish to update to. 7. System validates the new email address format. 8. If the new email address is invalid, the system displays an error message to the user. 9. If the new email address is valid, the system encounters an error while sending the verification OTP. 10. System displays an error message to the user, indicating the problem with sending the verification OTP. 11. User can choose to retry the email update process or contact support for assistance. 12. If the user chooses to retry, the process continues from Step 1. 13. If the user chooses to contact support, the process is temporarily halted until the issue is resolved. 14. Once the issue is resolved, the user can proceed with the email update process from the beginning. |

## FR04 Periodic Password Reset

|  |  |
| --- | --- |
| Req ID | FR04 |
| Title | Periodic Password Reset |
| Primary Actor | User |
| Pre-conditions | * User is logged in. * User's password has expired. |
| Scenario | 1. System prompts the user to enter a new password. 2. User enters a new password. 3. System validates the new password. 4. System updates the user's password in the system's database. 5. System logs the date and time of the password reset. 6. System sends a confirmation email to the user, confirming the password change. |
| Post-conditions | * User's password is successfully reset. * User receives a confirmation email. |
| Alternate Flow | 1. N/A |

## FR05 Incorrect Password Lock

|  |  |
| --- | --- |
| Req ID | FR05 |
| Title | Incorrect Password Lock |
| Primary Actor | User |
| Pre-conditions | * User has a registered account |
| Scenario | 1. User enters their username and password. 2. System verifies the entered credentials. 3. If the credentials are valid:    1. System resets the failed login attempt count to zero.    2. User is successfully logged in. 4. If the credentials are invalid:    1. System increments the failed login attempt count by one for the user.    2. If the failed login attempt count exceeds the predefined threshold:       1. System locks the user's account.       2. System displays an error message to the user indicating that their account is locked due to excessive failed login attempts. |
| Post-conditions | * User is successfully logged in. * User's account is locked if the failed login attempt threshold is exceeded. * User receives an error message if their account is locked |
| Alternate Flow | 4a. If the user's account is locked:   1. System displays an error message indicating that the account is locked. 2. System displays a countdown timer indicating the remaining lockout duration for the user. 3. User cannot access the account during the lockout duration, regardless of the password entered. |

## FR0X BlaBla

|  |  |
| --- | --- |
| Req ID |  |
| Title |  |
| Primary Actor |  |
| Pre-conditions |  |
| Scenario |  |
| Post-conditions |  |
| Alternate Flow |  |