Content

Create an Account (Any System User)

1. The system user fills the registration form in the browser.
2. The Account Web App validates the form data.
3. The Account Web App submits form data to the Account Microservice in the service layer.
4. The Account Microservice checks that the email is not already registered.
5. The Account Microservice encrypts the password field (and possibly other fields).
6. The Account Microservice saves the encrypted account data in the Account DB.
7. The Account Microservice composes an email message containing a verification URL which is generated based on the user's email address.
8. The verification email is saved in the Communication DB before it gets sent to the user's address via Communication Gateway.
9. The result of the transaction is returned back to the Account Microservice and then to the Account Web App.
10. The Account Web App will render the communication message and displays it back to the user.
11. The user opens his mailbox and clicks on the verification URL
12. The request will be submitted to the Account Web App then to the Account Microservice.
13. The Account Microservice verifies that the provided URL belongs to a registered but not activated account
14. The Account Microservice fetches the account data from Account DB and flags it as an activated account.
15. All read and write activities are logged using the Audit Microservice.

Reset Forgotten Password (Any System User)

1. System user submits his email address (his unique identifier) using a browser's web form.
2. The Account Web App ensures that the form contains a valid formatted email address.
3. The Account Web App submits the request to the Account Microservice.
4. The Account Microservice checks the Account DB to see if the account exists or not, and if does it fetches the account data.
5. The Account Microservice requests a temporary password from the Security Microservice
6. The Account Microservice updates the Account DB with the new password.
7. The Account Microservice composes the password reset email and embeds the temporary password within the message body.
8. The message is saved in the Communication DB and then sent to the user via Communication Gateway.
9. All read and write transactions are logged using the Audit Microservice.
10. The result of the transaction is returned to the Account Microservice then to the Account Web App.
11. The Account Web App renders the message and displays it to the user.
12. The user will receive an email with a temporary password and a URL that takes him to the Change Password Form.
13. The user fills the form data and submits it to the Account Web App which ensures that the new password and its confirmation match.
14. The form data is submitted to the Account Microservice.
15. The Account Microservice uses the security Microservice to authenticate the user with the temporary password and encrypts the new password.
16. The Account Microservice updates the Account DB with the new updated and encrypted password.
17. All read and write activities are logged using the Audit Microservice.
18. The result of the transaction is returned to the Account Microservice then to the Account Web App.
19. The Account Web App renders the message and displays it to the user.

Access and Update Profile (Birth Parent)

1. Birth Parents (BP) selects to access his profile, the request is submitted to the Profile Web App.
2. The Profile Web App uses the active session's credentials and passes them to the Profile Microservice to get the profile data.
3. The Profile Microservice first ensures that the BP has the right permissions to view the data via the Security Microservice.
4. The Profile Microservice requests the profile data from the Profile DB.
5. The Profile Microservice logs the profile access transactions using the Auditing Microservice.
6. The Profile Microservice returns back the profile data to the Profile Web App.
7. The Profile Web App renders and formats the profile data and displaying them to the BP.
8. Updating BP profiles is exactly the same with differences only in the permission and the transaction type.

Send Secured Message

1. System Users can send and receive messages in a secured environment.

8. Updating BP profiles is exactly the same with differences only in the permission and the transaction type.

Send Secured Message
(Any System User)

1. System Users can send and receive messages in a secured environment.
2. The user fills in a message form in his browser and submits it.
3. Message data will be encapsulated in a document (message) and sent from the Communication Web App to the Communication Microservice.
4. The Communication Microservice authenticates the user through the Security Microservice.
5. The Security Microservice encrypts the message and returns it back to the Communication Microservice.
6. The Communication Microservice saves the message in the Communication DB.
7. The Communication Microservice composes the notification message to be sent to the user's email (or phone in case of SMS).
8. The Communication Gateway sends the notification message to the user.
9. The result of the transaction is logged for auditing and returned back to the Communication Web App.
10. The Communication Wen App renders and formats the result and displays it to the user.

Read Secured Message
(Any System User)

1. The system user will be notified by an email (or other communication method) that a secured message has been received.
2. The system user selects to view his secured inbox, and the request is sent to Communication Web App.
3. The Communication Web App requests all message headers in the inbox for the active session's user from the Communication Microservice.
4. The Communication Microservice uses the Security Microservice to authenticate the user and checks his permissions.
5. The Communication Microservice requests message headers from the Communication DB.
6. The results is returned to the Communication Web App which renders, formats the data, and display it to the user.
7. When the user clicks on a specific message to read it, the request is sent to the Communication Web App with the message ID.
8. The Communication Web App forwards the request to the Communication Microservice.
9. The Communication Microservice uses the Security Microservice to authenticate the user and check his permissions.
10. The Communication Microservice fetches the message data from the Communication DB and updates its status to “Read”.
11. All read and write transactions will be logged for auditing using the Auditing Microservice.
12. Messages in the Communication DB are encrypted so the message data has to be decrypted.
13. The Communication Microservice uses the Security Microservice to decrypt the message and returns the result to the Communication Web App.
14. The Communication Web App renders the message and displays to the user.

Access and Update a Plan
(Birth Parent)

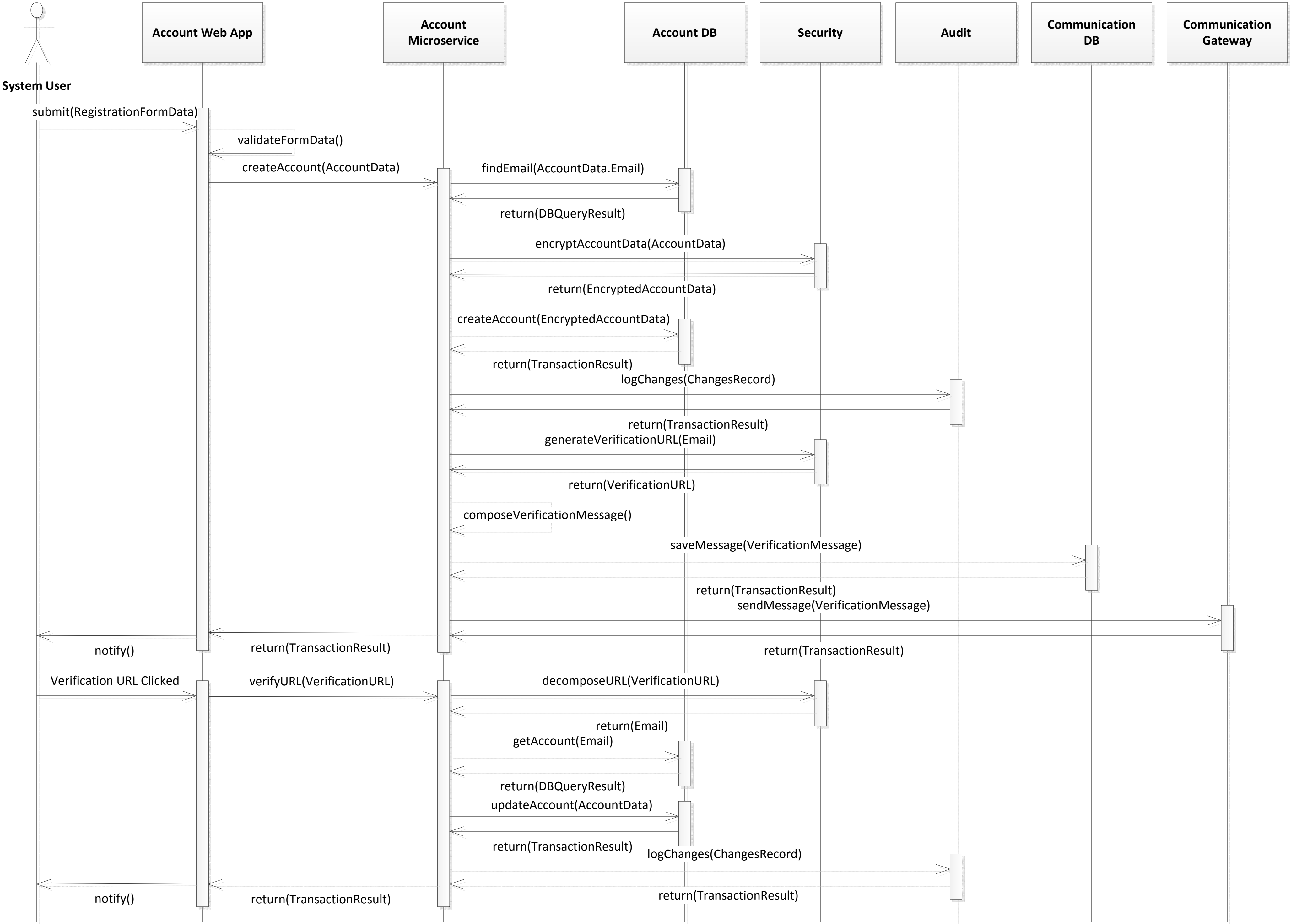
1. The Birth Parents (BP) can access and update their plans.
2. The BP selects to access his provided plan, and his request is submitted to the Plan Web App.
3. The Plan Web App uses the active session's credentials and passes them to the Plan Microservice to get the plan data.
4. The Plan Microservice uses Security Microservice to ensure that the BP has the right permissions to view plan’s data.
5. The Plan Microservice requests the plan data from Plan DB.
6. The Plan Microservice uses the Auditing Microservice to log the access transaction.
7. The Plan Microservice returns the plan data to the Plan Web App.
8. The Plan Web App render and format the plan data then displays them to the BP.
9. Updating plans by BP is exactly the same with differences in the permission type and the transaction type.

Search Facilities
(Birth Parent)

1. The Birth Parent (BP) has the ability to search for caring facilities within their zip code.
2. BP enters the value of the zip code and submits it to the Facility Web App.
3. The Facility Web App uses the active session's credentials and submits them along with the zip code to the Facility Microservice.
4. The Facility Microservice asks the Security Microservice to authenticate the user and making sure that he has the permissions to do the search.
5. The Facility Microservice queries the Facility Open Data DB and gets the search results.
6. The Facility Microservice returns the results to the Facility Web App.
7. The Facility Web App formats and displays the results to the BP.

Create an Account Sequence:

← Home


System User
Account Web App
Account Microservice
Account DB
Security
Audit
Communication DB
Communication Gateway
submit(RegistrationFormData)
validateFormData()
createAccount(AccountData)
findEmail(AccountData.Email)
return(DBQueryResult)
encryptAccountData(AccountData)
return(EncryptedAccountData)
createAccount(EncryptedAccountData)
return(TransactionResult)
logChanges(ChangesRecord)
return(TransactionResult)
generateVerificationURL(Email)
return(VerificationURL)
composeVerificationMessage()
saveMessage(VerificationMessage)
return(TransactionResult)
sendMessage(VerificationMessage)
notify()
return(TransactionResult)
return(TransactionResult)
Verification URL Clicked
verifyURL(VerificationURL)
decomposeURL(VerificationURL)
return(Email)
getAccount(Email)
return(DBQueryResult)
updateAccount(AccountData)
return(TransactionResult)
logChanges(ChangesRecord)
notify()
return(TransactionResult)
return(TransactionResult)

Create an Account Classes:

AccountFormData

- Email
- Password
- Confirmed Password

AccountData

- Email
- Password
- Activated?

EncryptedAccountData

- Email
- EncryptedPassword

DBQueryResult

- QueryResultsCollection

TransactionResult

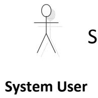
- ResultsCollection

ChangesRecord

- Email //UserID
- OldValue
- NewValue
- TimeStamp

VerificationMessage

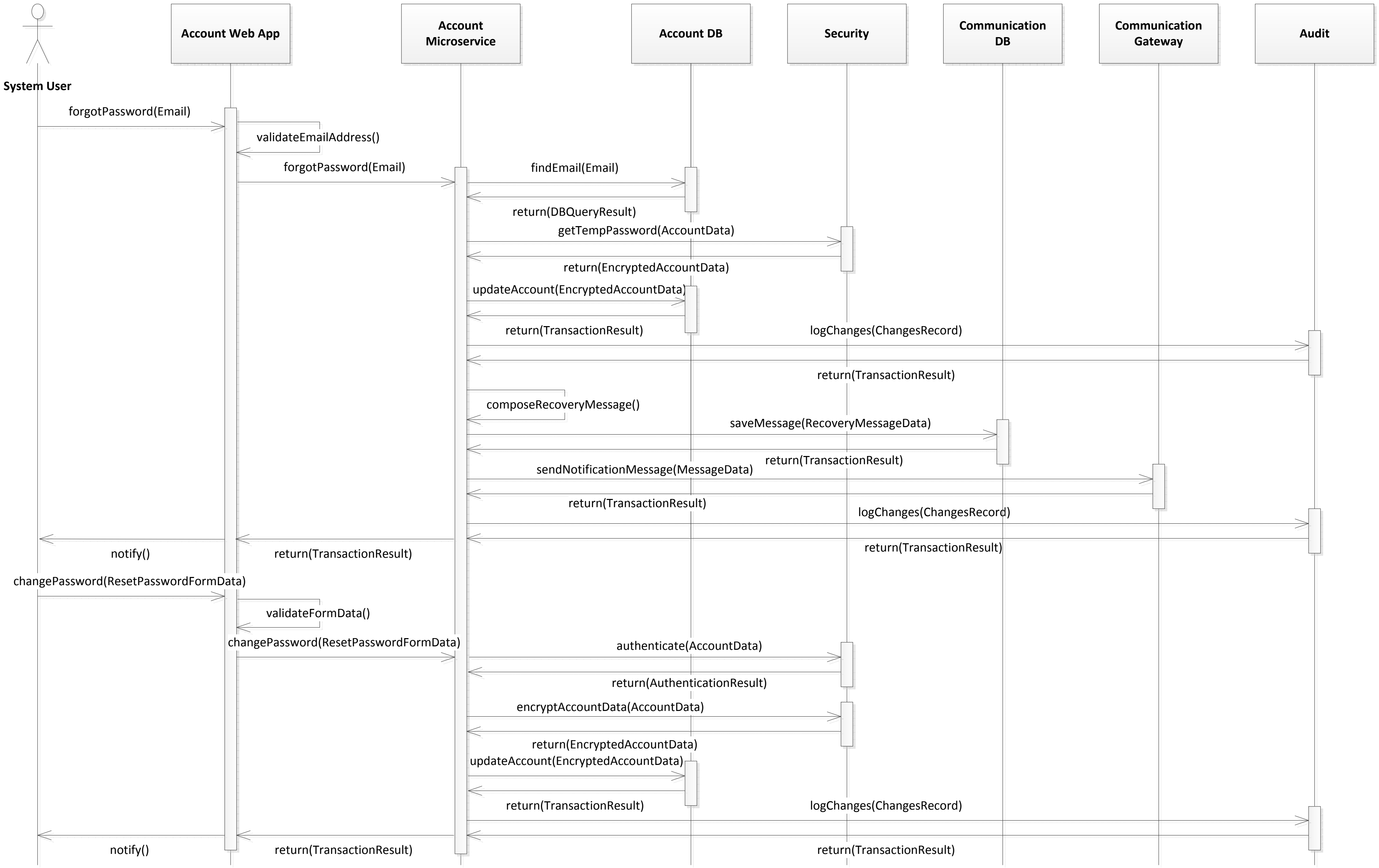
- SenderEmail
- ReceiverEmail
- Subject
- Body
- VerificationURL



System User can be a Birth Parent, a Foster Parent, or a Case Worker

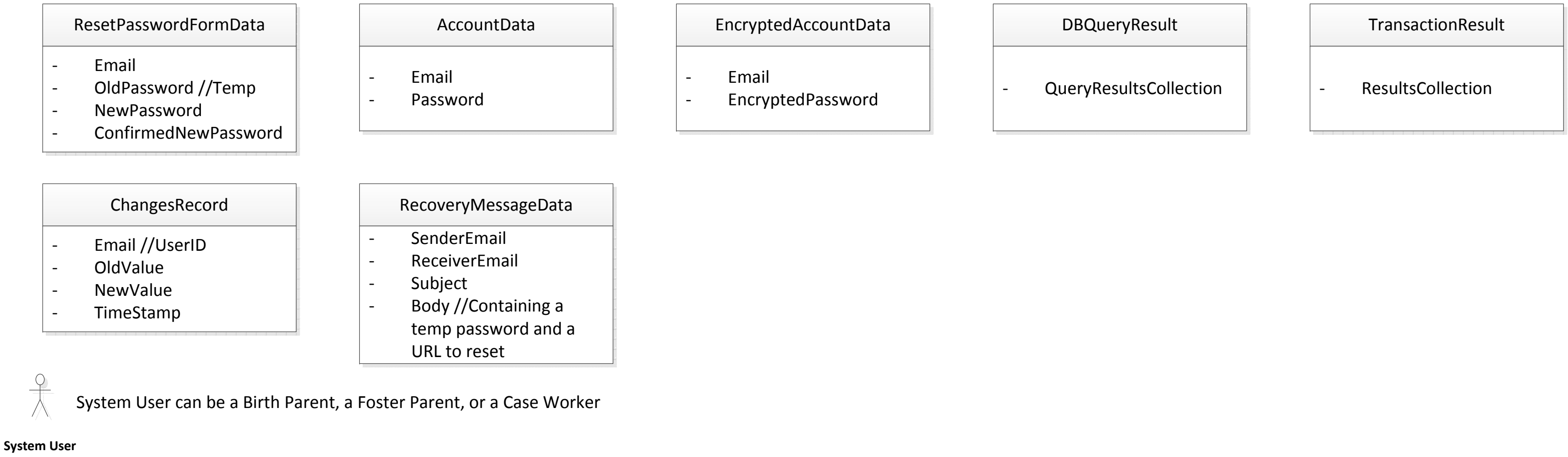
Reset Password Sequence:

← Home


System User
Account Web App
Account Microservice
Account DB
Security
Communication DB
Communication Gateway
Audit
forgotPassword(Email)
validateEmailAddress()
forgotPassword(Email)
findEmail(Email)
return(DBQueryResult)
getTempPassword(AccountData)
return(EncryptedAccountData)
updateAccount(EncryptedAccountData)
return(TransactionResult)
logChanges(ChangesRecord)
return(TransactionResult)
composeRecoveryMessage()
saveMessage(RecoveryMessageData)
return(TransactionResult)
sendNotificationMessage(MessageData)
return(TransactionResult)
logChanges(ChangesRecord)
return(TransactionResult)
notify()
return(TransactionResult)
changePassword(ResetPasswordFormData)
validateFormData()
changePassword(ResetPasswordFormData)
authenticate(AccountData)
return(AuthenticationResult)
encryptAccountData(AccountData)
return(EncryptedAccountData)
updateAccount(EncryptedAccountData)
return(TransactionResult)
logChanges(ChangesRecord)
notify()
return(TransactionResult)
return(TransactionResult)

Reset Password Classes:

Reset Password Classes:



ResetPasswordFormData

- Email
- OldPassword //Temp
- NewPassword
- ConfirmedNewPassword

AccountData

- Email
- Password

EncryptedAccountData

- Email
- EncryptedPassword

DBQueryResult

- QueryResultsCollection

TransactionResult

- ResultsCollection

ChangesRecord

- Email //UserID
- OldValue
- NewValue
- TimeStamp

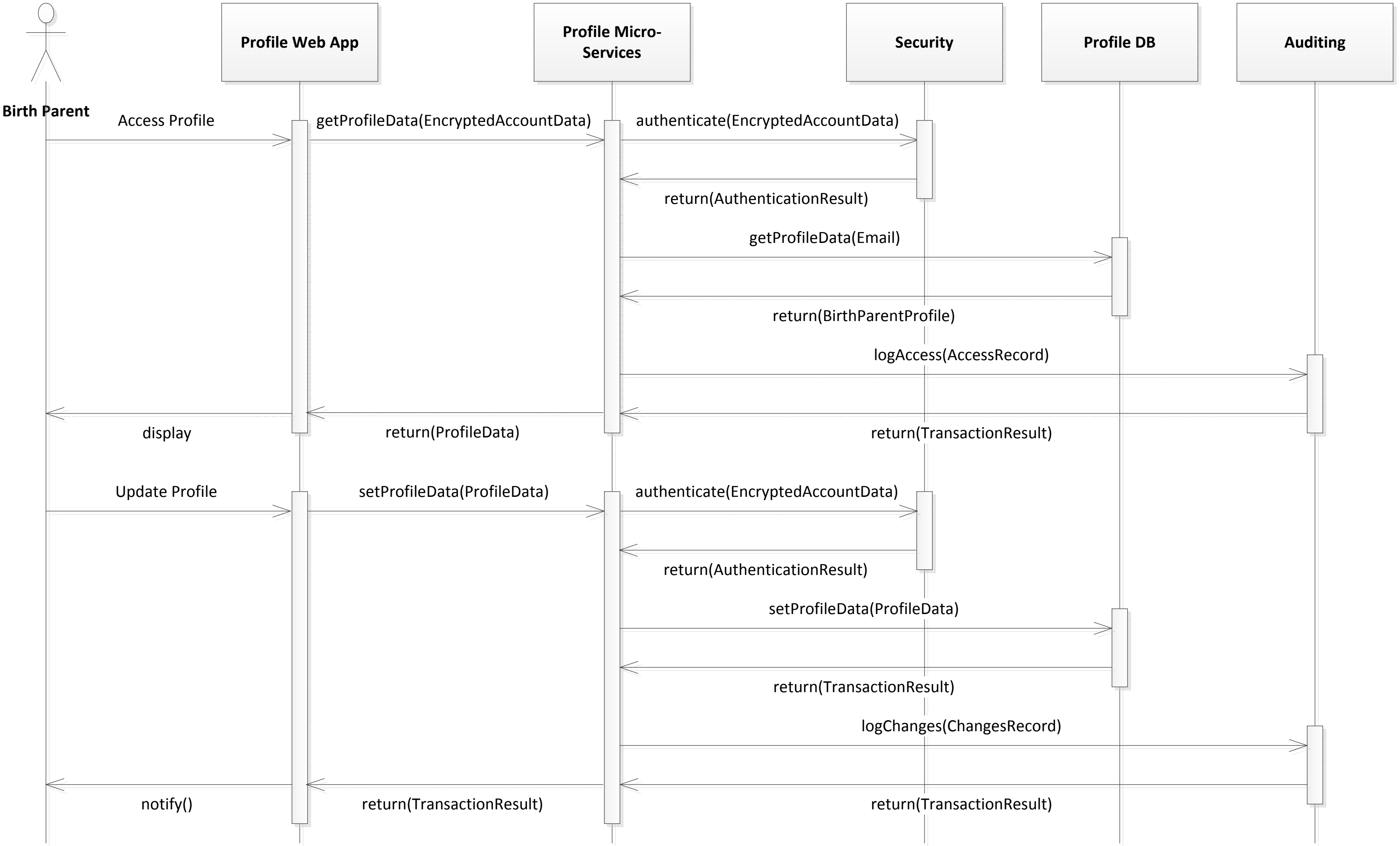
RecoveryMessageData

- SenderEmail
- ReceiverEmail
- Subject
- Body //Containing a temp password and a URL to reset

System User can be a Birth Parent, a Foster Parent, or a Case Worker

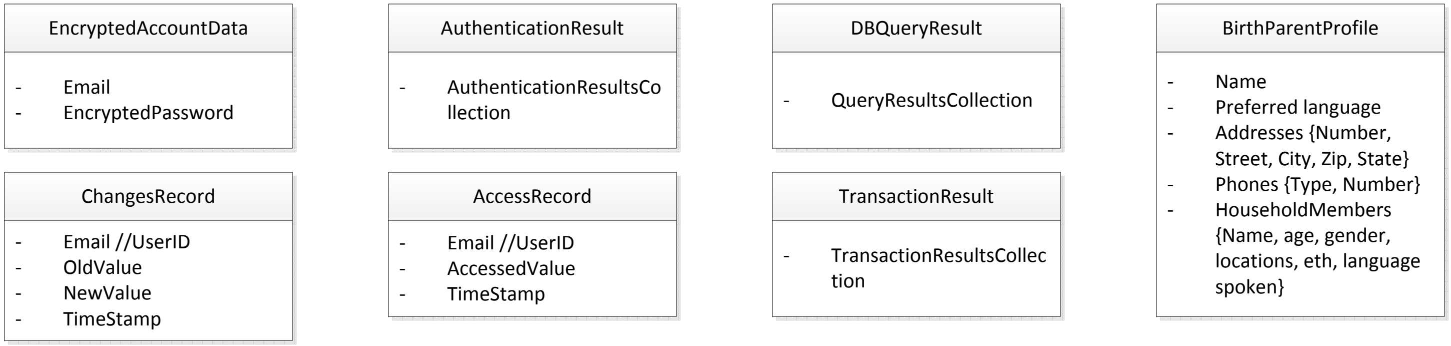
Access and Update Profile Sequence:

← Home


Birth Parent
Profile Web App
Profile Micro-Services
Security
Profile DB
Auditing
Access Profile
getProfileData(EncryptedAccountData)
authenticate(EncryptedAccountData)
return(AuthenticationResult)
getProfileData(Email)
return(BirthParentProfile)
logAccess(AccessRecord)
display
return(ProfileData)
return(TransactionResult)
Update Profile
setProfileData(ProfileData)
authenticate(EncryptedAccountData)
return(AuthenticationResult)
setProfileData(ProfileData)
return(TransactionResult)
logChanges(ChangesRecord)
notify()
return(TransactionResult)
return(TransactionResult)

Access and Update Profile Classes:

Access and Update Profile Classes:



EncryptedAccountData

- Email
- EncryptedPassword

ChangesRecord

- Email //UserID
- OldValue
- NewValue
- TimeStamp

AuthenticationResult

- AuthenticationResultsCollection

AccessRecord

- Email //UserID
- AccessedValue
- TimeStamp

DBQueryResult

- QueryResultsCollection

TransactionResult

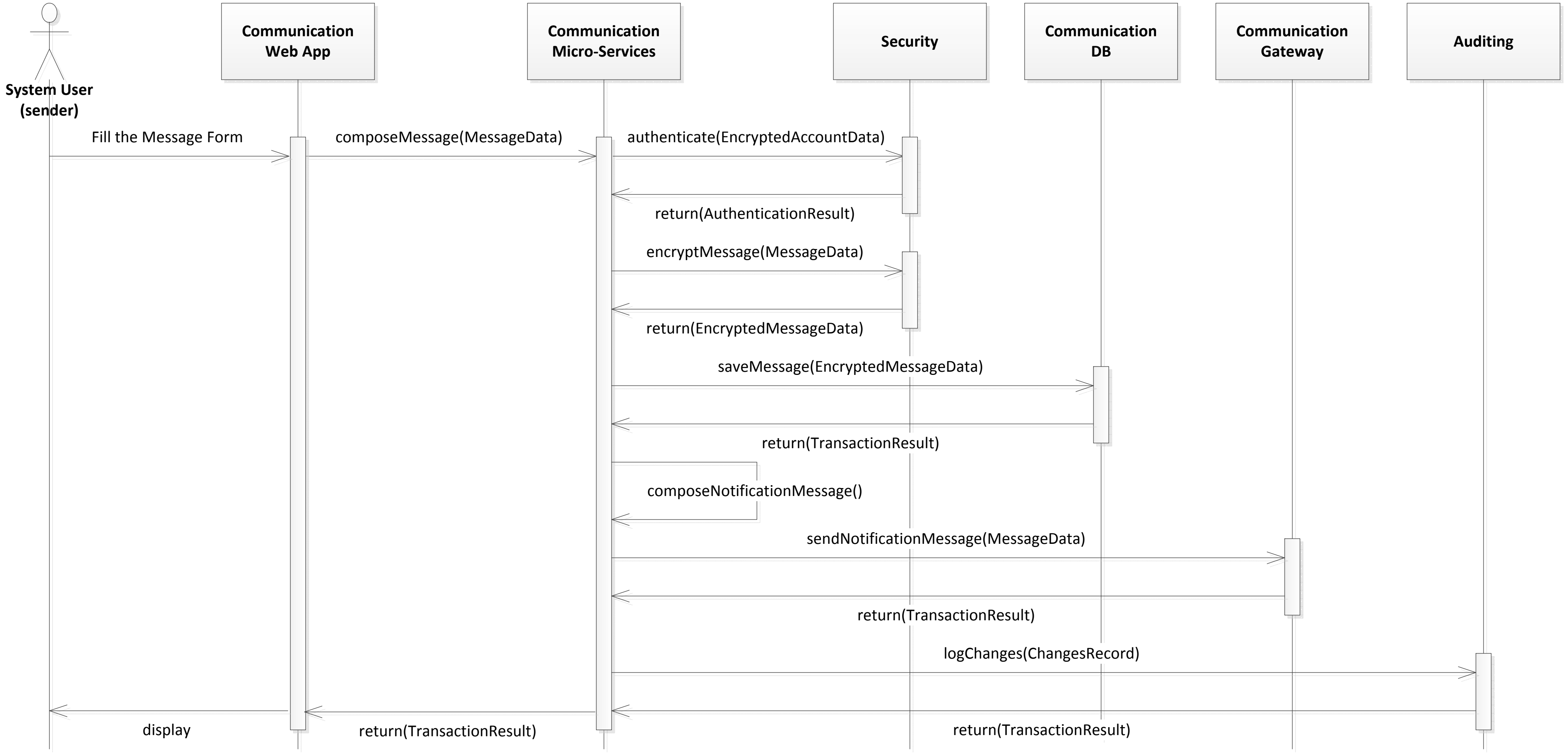
- TransactionResultsCollection

BirthParentProfile

- Name
- Preferred language
- Addresses {Number, Street, City, Zip, State}
- Phones {Type, Number}
- HouseholdMembers {Name, age, gender, locations, eth, language spoken}

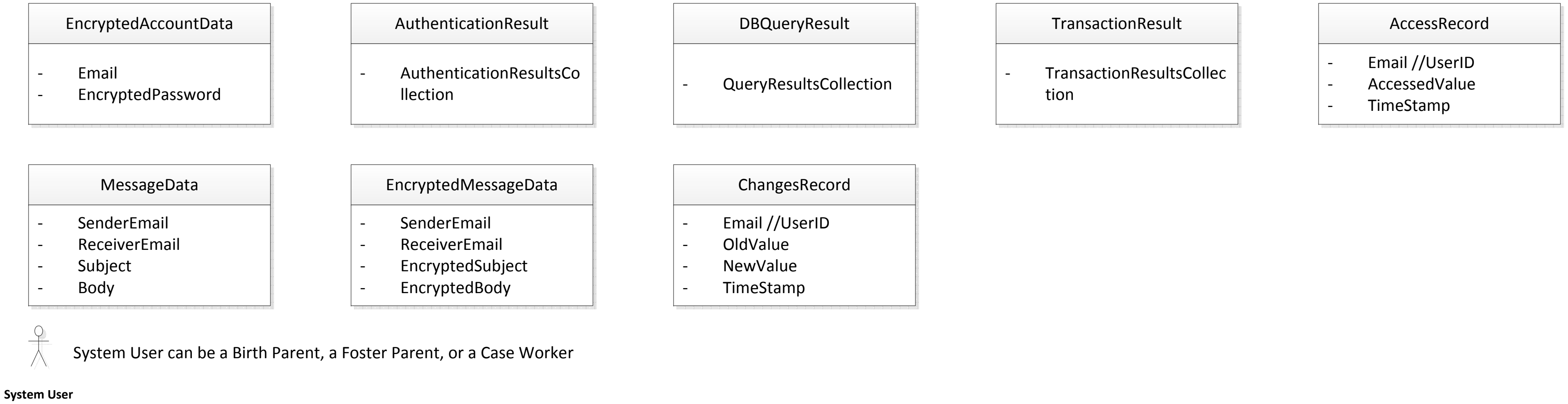
Send Message Sequence:

← Home


System User (sender)
Communication Web App
Communication Micro-Services
Security
Communication DB
Communication Gateway
Auditing
Fill the Message Form
composeMessage(MessageData)
authenticate(EncryptedAccountData)
return(AuthenticationResult)
encryptMessage(MessageData)
return(EncryptedMessageData)
saveMessage(EncryptedMessageData)
return(TransactionResult)
composeNotificationMessage()
sendNotificationMessage(MessageData)
return(TransactionResult)
logChanges(ChangesRecord)
display
return(TransactionResult)
return(TransactionResult)

Send Message Classes:

Send Message Classes:



EncryptedAccountData

- Email
- EncryptedPassword

AuthenticationResult

- AuthenticationResultsCollection

DBQueryResult

- QueryResultsCollection

TransactionResult

- TransactionResultsCollection

AccessRecord

- Email //UserID
- AccessedValue
- TimeStamp

MessageData

- SenderEmail
- ReceiverEmail
- Subject
- Body

EncryptedMessageData

- SenderEmail
- ReceiverEmail
- EncryptedSubject
- EncryptedBody

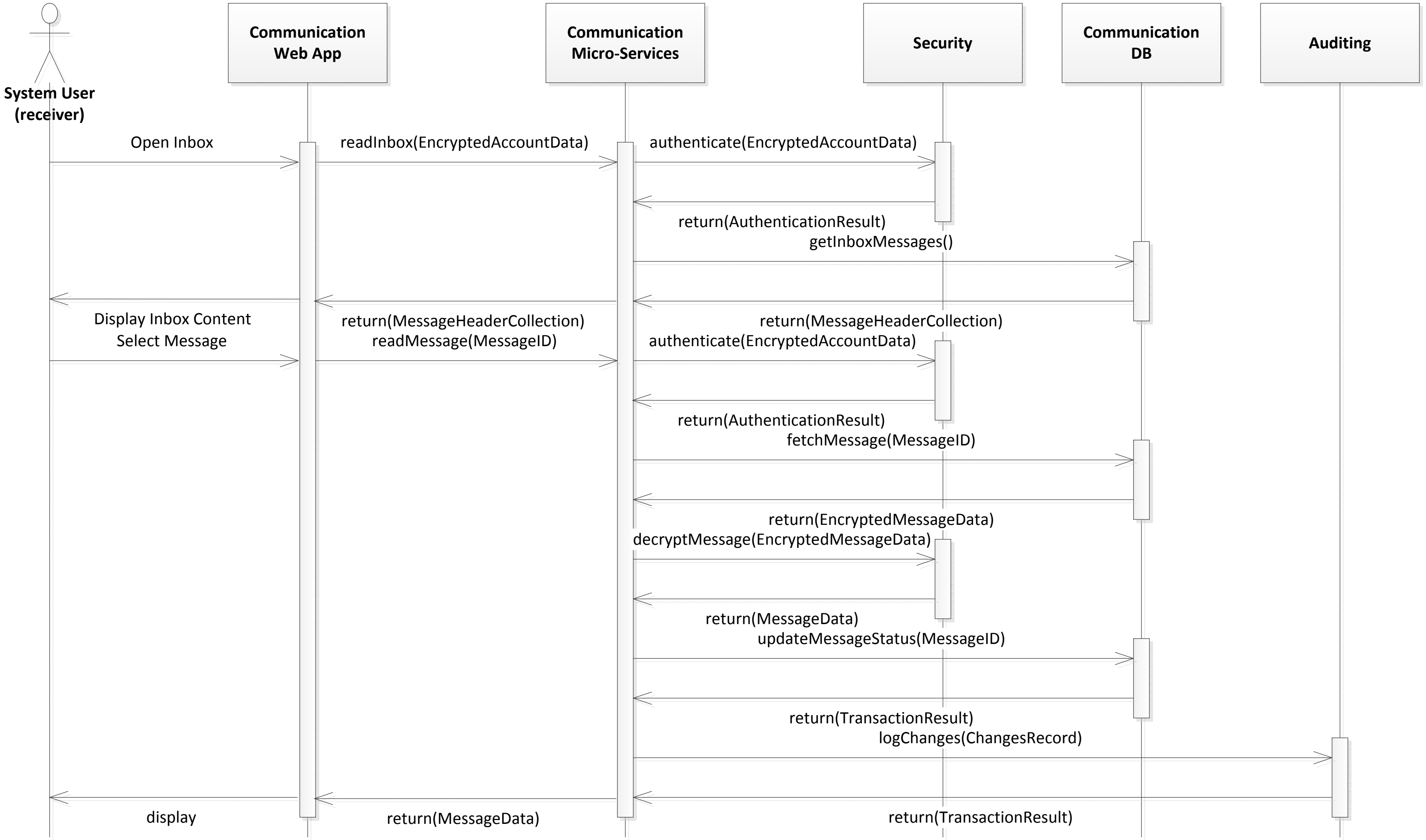
ChangesRecord

- Email //UserID
- OldValue
- NewValue
- TimeStamp

System User can be a Birth Parent, a Foster Parent, or a Case Worker

Read Message Sequence:

← Home


System User (receiver)
Communication Web App
Communication Micro-Services
Security
Communication DB
Auditing
Open Inbox
readInbox(EncryptedAccountData)
authenticate(EncryptedAccountData)
return(AuthenticationResult)
getInboxMessages()
Display Inbox Content
return(MessageHeaderCollection)
return(MessageHeaderCollection)
Select Message
readMessage(MessageID)
authenticate(EncryptedAccountData)
return(AuthenticationResult)
fetchMessage(MessageID)
return(EncryptedMessageData)
decryptMessage(EncryptedMessageData)
return(MessageData)
updateMessageStatus(MessageID)
return(TransactionResult)
logChanges(ChangesRecord)
display
return(MessageData)
return(TransactionResult)

Read Message Classes:

Read Message Classes:

EncryptedAccountData

- Email
- EncryptedPassword

AuthenticationResult

- AuthenticationResultsCollection

DBQueryResult

- QueryResultsCollection

TransactionResult

- TransactionResultsCollection

AccessRecord

- Email //UserID
- AccessedValue
- TimeStamp

MessageData

- SenderEmail
- ReceiverEmail
- DateTimeStamp
- Subject
- Body
- Status

EncryptedMessageData

- SenderEmail
- ReceiverEmail
- DateTimeStamp
- EncryptedSubject
- EncryptedBody
- Status

ChangesRecord

- Email //UserID
- OldValue
- NewValue
- TimeStamp

MessageHeader

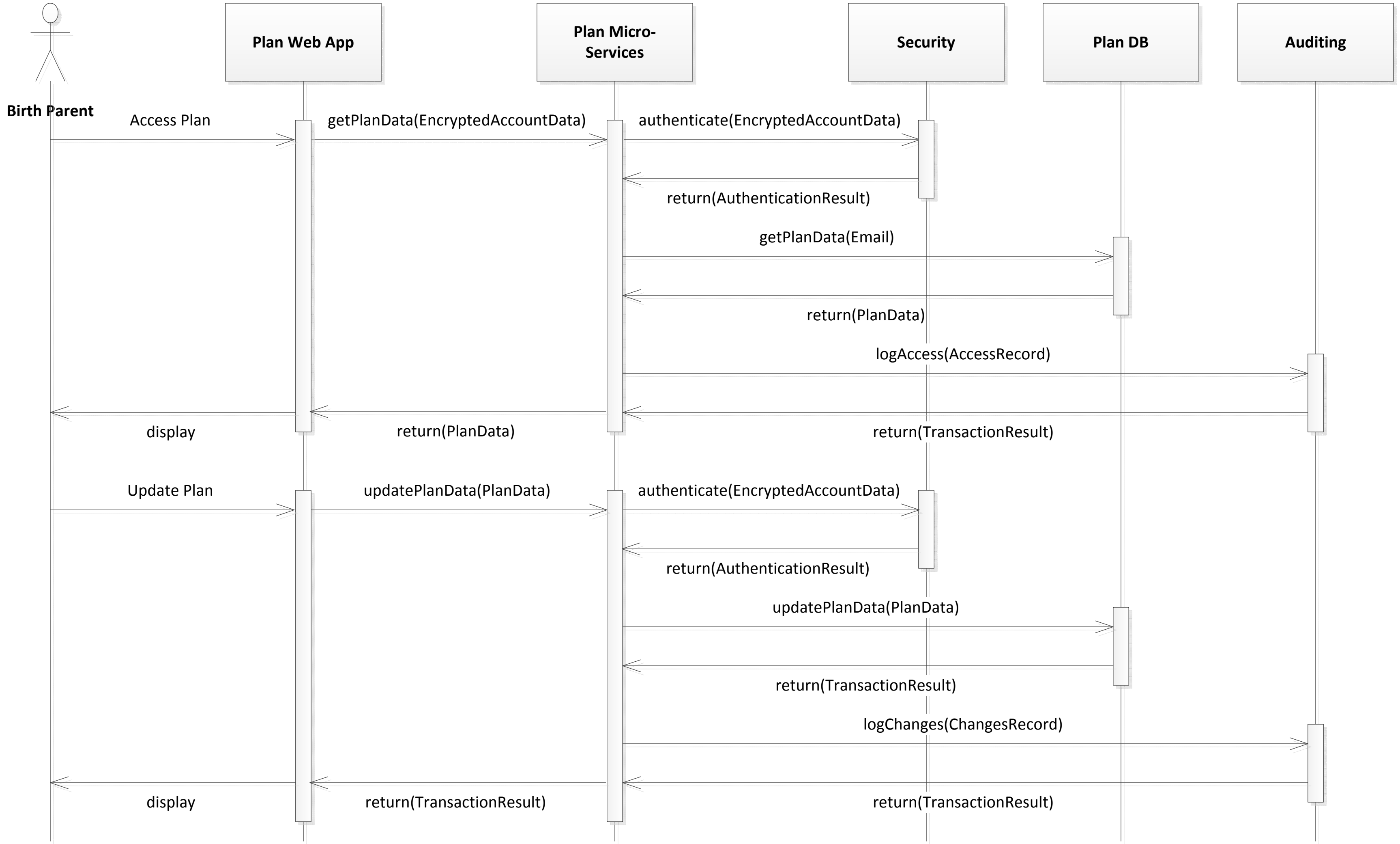
- SenderEmail
- DateTime
- Subject
- Status

System User can be a Birth Parent, a Foster Parent, or a Case Worker

System User

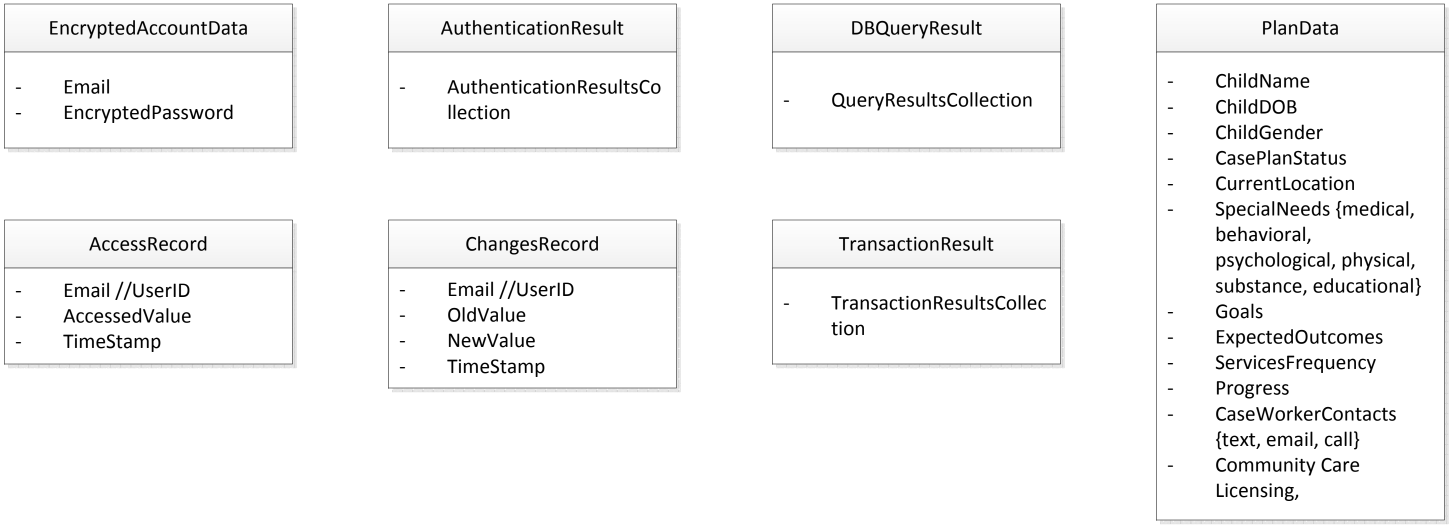
Access and Update Plan Sequence:

← Home


Birth Parent
Plan Web App
Plan Micro-Services
Security
Plan DB
Auditing
Access Plan
getPlanData(EncryptedAccountData)
authenticate(EncryptedAccountData)
return(AuthenticationResult)
getPlanData(Email)
return(PlanData)
logAccess(AccessRecord)
display
return(PlanData)
return(TransactionResult)
Update Plan
updatePlanData(PlanData)
authenticate(EncryptedAccountData)
return(AuthenticationResult)
updatePlanData(PlanData)
return(TransactionResult)
logChanges(ChangesRecord)
display
return(TransactionResult)
return(TransactionResult)

Access and Update Plan Classes:

Access and Update Plan Classes:



EncryptedAccountData

- Email
- EncryptedPassword

AuthenticationResult

- AuthenticationResultsCollection

DBQueryResult

- QueryResultsCollection

PlanData

- ChildName
- ChildDOB
- ChildGender
- CasePlanStatus
- CurrentLocation
- SpecialNeeds {medical, behavioral, psychological, physical, substance, educational}
- Goals
- ExpectedOutcomes
- ServicesFrequency
- Progress
- CaseWorkerContacts {text, email, call}
- Community Care Licensing,

AccessRecord

- Email //UserID
- AccessedValue
- TimeStamp

ChangesRecord

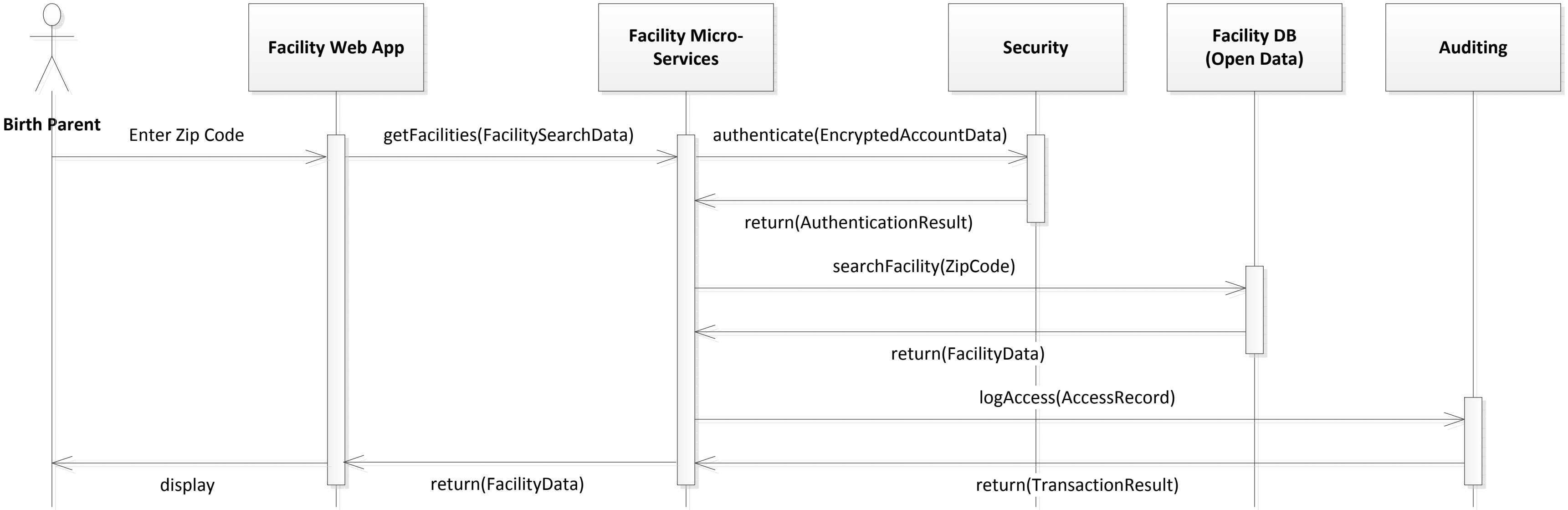
- Email //UserID
- OldValue
- NewValue
- TimeStamp

TransactionResult

- TransactionResultsCollection

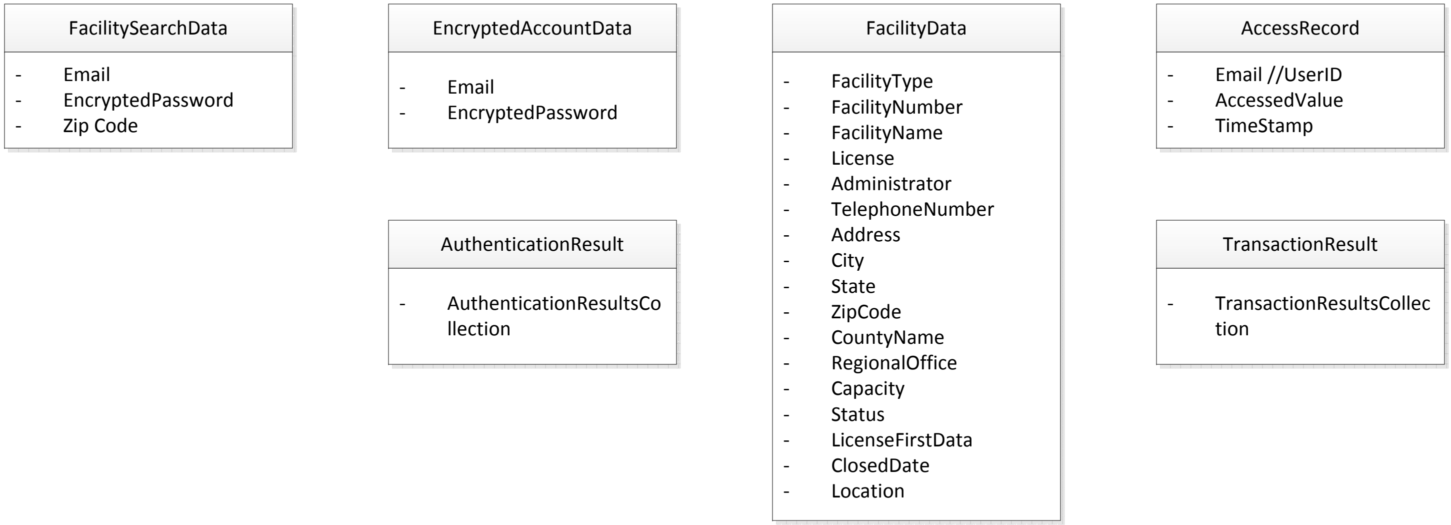
Find Facility Sequence:

← Home



Find Facility Classes:

Find Facility Classes:



FacilitySearchData

- Email
- EncryptedPassword
- Zip Code

EncryptedAccountData

- Email
- EncryptedPassword

AuthenticationResult

- AuthenticationResultsCollection

FacilityData

- FacilityType
- FacilityNumber
- FacilityName
- License
- Administrator
- TelephoneNumber
- Address
- City
- State
- ZipCode
- CountyName
- RegionalOffice
- Capacity
- Status
- LicenseFirstData
- ClosedDate
- Location

AccessRecord

- Email //UserID
- AccessedValue
- TimeStamp

TransactionResult

- TransactionResultsCollection