MIDDLE EAST TECHNICAL UNIVERSITY

NORTHERN CYPRUS CAMPUS

Software Engineering - CNG 350
Semester Project Design

Deniz Yılmazok

Notes

Given date examples in this section are in the format Day/Month/Year with 24 hour period.

- Our design for reminders is kept very simple. Reminders just store Date object and integer named interval to be notified. For example, let's assume we have a reminder object with Date of 01/01/2017 15:00 and interval which is 3. Then, this reminder will be reminded by the system as follows:

  - 04/01/2017 15:00
  - 07/01/2017 15:00
  - 10/01/2017 15:00

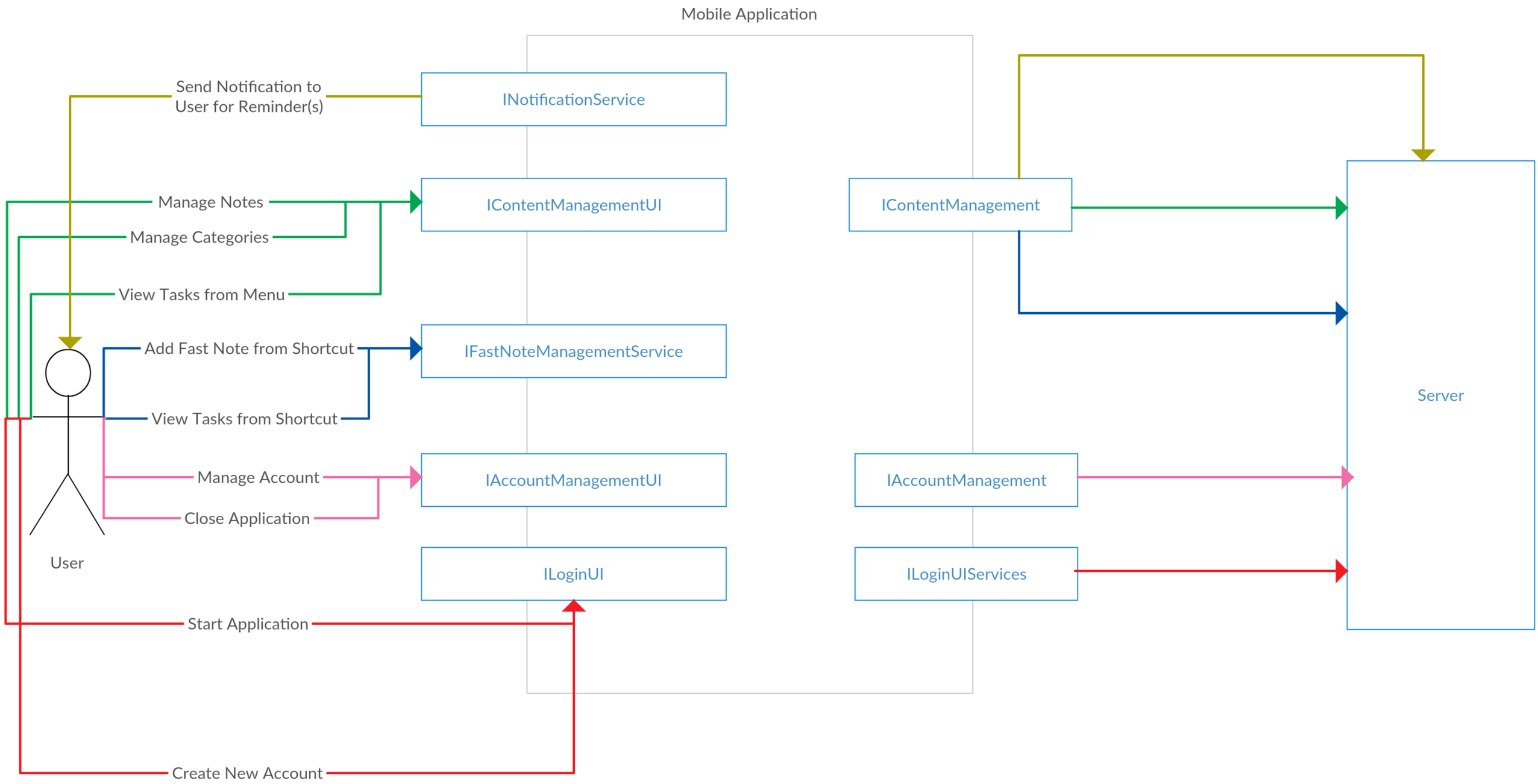
Notes:

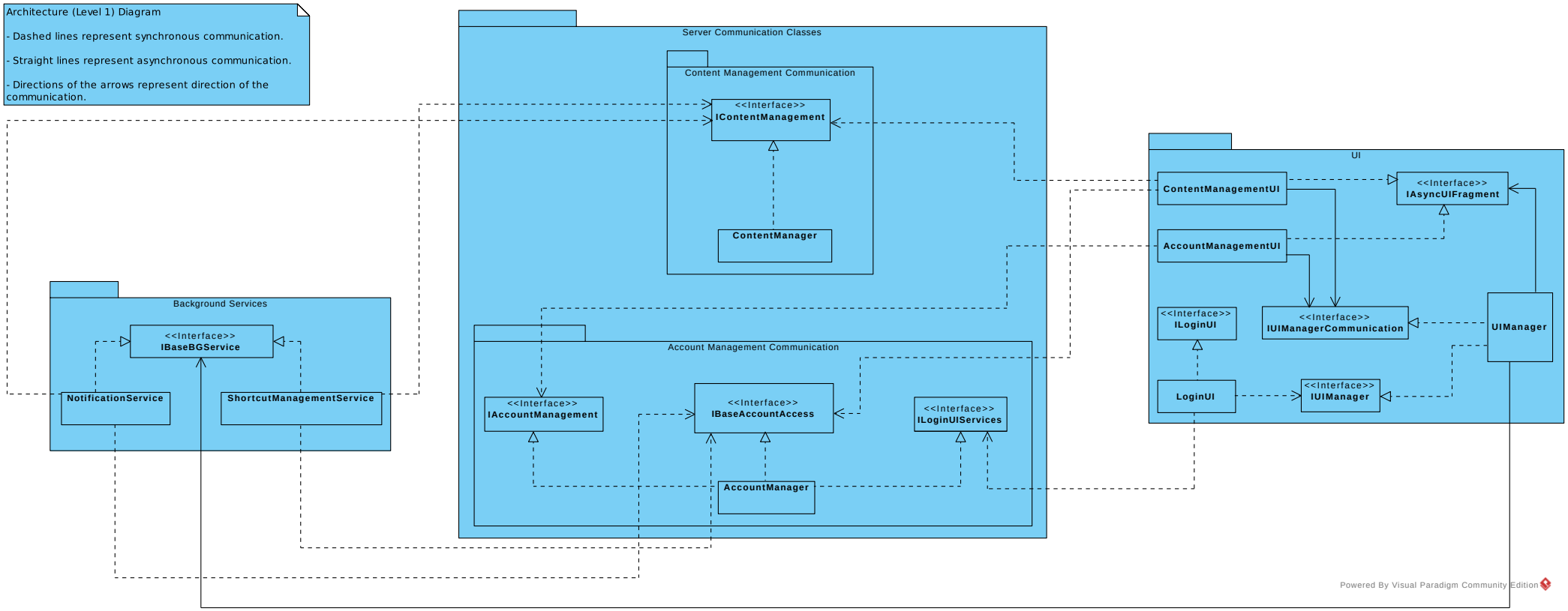
- Arrow from IFastNoteManagementService indicates notification is sent to user by system. That is, interaction is started by system. After that, user is able to interact with the notification.

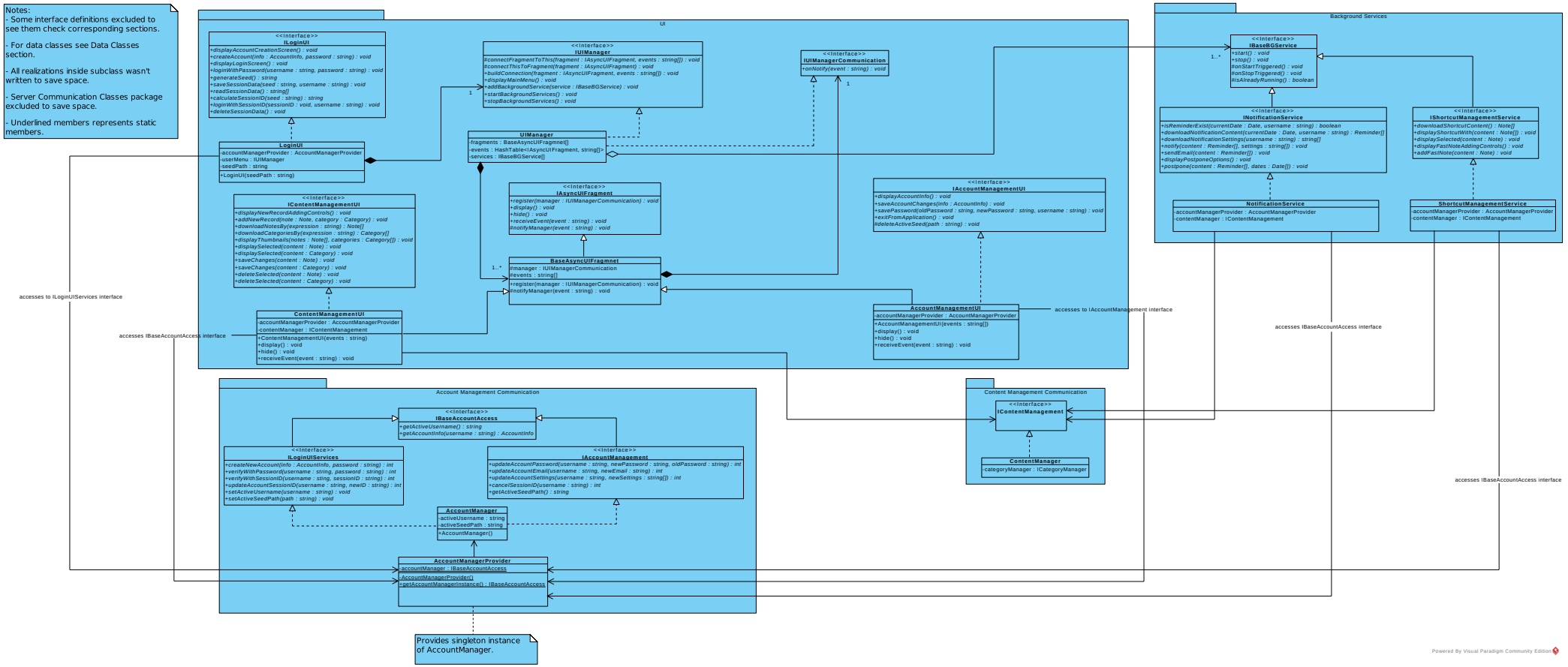
- IAccountManagement and ILoginUIServices could be merged as a one interface. But, they are separate from each other because of the security reasons.

- IContentManagement, IAccountManagement and ILoginUIServices are implemented inside the mobile application and they use APIs provided on server side for use cases represented with corresponding colors.

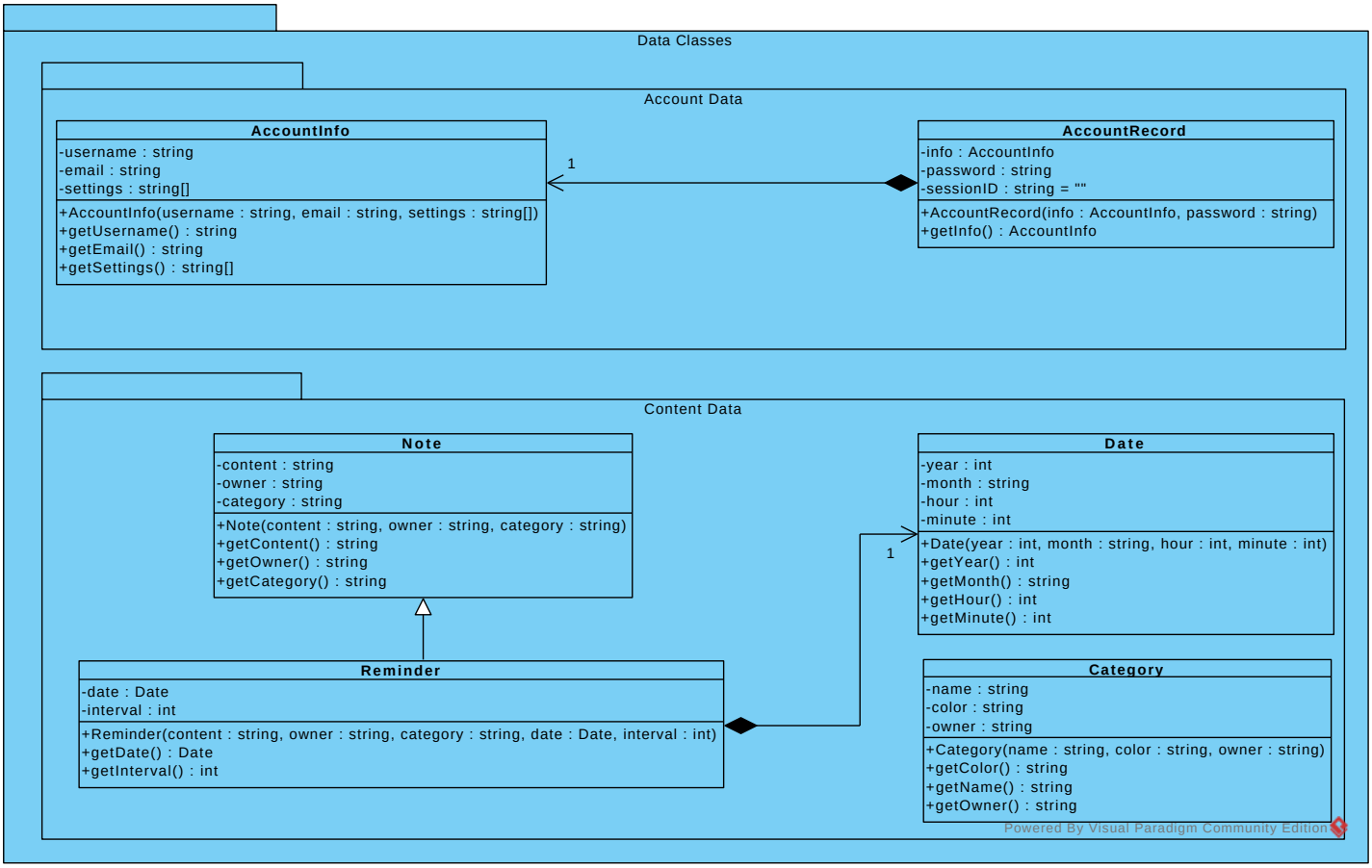
Context (Level 0) Diagram




Architecture (Level 1) Diagram
- Dashed lines represent synchronous communication.
- Straight lines represent asynchronous communication.
- Directions of the arrows represent direction of the communication.
Server Communication Classes
Content Management Communication
<<Interface>>
IContentManagement
ContentManager
Account Management Communication
<<Interface>>
IAccountManagement
<<Interface>>
IBaseAccountAccess
<<Interface>>
ILoginUIServices
AccountManager
Background Services
<<Interface>>
IBaseBGService
NotificationService
ShortcutManagementService
UI
ContentManagementUI
<<Interface>>
IAsyncUIFragment
AccountManagementUI
<<Interface>>
ILoginUI
<<Interface>>
IUIManagerCommunication
UIManager
LoginUI
<<Interface>>
IUIManager
Powered By Visual Paradigm Community Edition


Notes:
- Some interface definitions excluded to see them check corresponding sections.
- For data classes see Data Classes section.
- All realizations inside subclass wasn't written to save space.
- Server Communication Classes package excluded to save space.
- Underlined members represents static members.
UI
<<Interface>>
ILoginUI
+displayAccountCreationScreen() : void
+createAccount(info : AccountInfo, password : string) : void
+displayLoginScreen() : void
+loginWithPassword(username : string, password : string) : void
+generateSeed() : string
+saveSessionData(seed : string, username : string) : void
+readSessionData() : string[]
+calculateSessionID(seed : string) : string
+loginWithSessionID(sessionID : void, username : string) : void
+deleteSessionData() : void
LoginUI
-accountManagerProvider : AccountManagerProvider
-userMenu : IUIManager
-seedPath : string
+LoginUI(seedPath : string)
<<Interface>>
IUIManager
#connectFragmentToThis(fragment : IAsyncUIFragment, events : string[]) : void
#connectThisToFragment(fragment : IAsyncUIFragment) : void
+buildConnection(fragment : IAsyncUIFragment, events : string[]) : void
+displayMainMenu() : void
+addBackgroundService(service : IBaseBGService) : void
+startBackgroundServices() : void
+stopBackgroundServices() : void
<<Interface>>
IUIManagerCommunication
+onNotify(event : string) : void
UIManager
-fragments : BaseAsyncUIFragmnet[]
-events : HashTable<IAsyncUIFragment, string[]>
-services : IBaseBGService[]
<<Interface>>
IContentManagementUI
+displayNewRecordAddingControls() : void
+addNewRecord(note : Note, category : Category) : void
+downloadNotesBy(expression : string) : Note[]
+downloadCategoriesBy(expression : string) : Category[]
+displayThumbnails(notes : Note[], categories : Category[]) : void
+displaySelected(content : Note) : void
+displaySelected(content : Category) : void
+saveChanges(content : Note) : void
+saveChanges(content : Category) : void
+deleteSelected(content : Note) : void
+deleteSelected(content : Category) : void
<<Interface>>
IAsyncUIFragment
+register(manager : IUIManagerCommunication) : void
+display() : void
+hide() : void
+receiveEvent(event : string) : void
#notifyManager(event : string) : void
BaseAsyncUIFragmnet
#manager : IUIManagerCommunication
#events : string[]
+register(manager : IUIManagerCommunication) : void
#notifyManager(event : string) : void
<<Interface>>
IAccountManagementUI
+displayAccountInfo() : void
+saveAccountChanges(info : AccountInfo) : void
+savePassword(oldPassword : string, newPassword : string, username : string) : void
+exitFromApplication() : void
#deleteActiveSeed(path : string) : void
ContentManagementUI
-accountManagerProvider : AccountManagerProvider
-contentManager : IContentManagement
+ContentManagementUI(events : string)
+display() : void
+hide() : void
+receiveEvent(event : string) : void
AccountManagementUI
-accountManagerProvider : AccountManagerProvider
+AccountManagementUI(events : string[])
+display() : void
+hide() : void
+receiveEvent(event : string) : void
accesses to ILoginUIServices interface
accesses IBaseAccountAccess interface
accesses to IAccountManagement interface
Background Services
<<Interface>>
IBaseBGService
+start() : void
+stop() : void
#onStartTriggered() : void
#onStopTriggered() : void
#isAlreadyRunning() : boolean
1..*
<<Interface>>
INotificationService
+isReminderExist(currentDate : Date, username : string) : boolean
+downloadNotificationContent(currentDate : Date, username : string) : Reminder[]
+downloadNotificationSettings(username : string) : string[]
+notify(content : Reminder[], settings : string[]) : void
+sendEmail(content : Reminder[]) : void
+displayPostponeOptions() : void
+postpone(content : Reminder[], dates : Date[]) : void
<<Interface>>
IShortcutManagementService
+downloadShortcutContent() : Note[]
+displayShortcutWith(content : Note[]) : void
+displaySelected(content : Note) : void
+displayFastNoteAddingControls() : void
+addFastNote(content : Note) : void
NotificationService
-accountManagerProvider : AccountManagerProvider
-contentManager : IContentManagement
ShortcutManagementService
-accountManagerProvider : AccountManagerProvider
-contentManager : IContentManagement
accesses IBaseAccountAccess interface
Account Management Communication
<<Interface>>
IBaseAccountAccess
+getActiveUsername() : string
+getAccountInfo(username : string) : AccountInfo
<<Interface>>
ILoginUIServices
+createNewAccount(info : AccountInfo, password : string) : int
+verifyWithPassword(username : string, password : string) : int
+verifyWithSessionID(username : string, sessionID : string) : int
+updateAccountSessionID(username : string, newID : string) : int
+setActiveUsername(username : string) : void
+setActiveSeedPath(path : string) : void
<<Interface>>
IAccountManagement
+updateAccountPassword(username : string, newPassword : string, oldPassword : string) : int
+updateAccountEmail(username : string, newEmail : string) : int
+updateAccountSettings(username : string, newSettings : string[]) : int
+cancelSessionID(username : string) : int
+getActiveSeedPath() : string
AccountManager
-activeUsername : string
-activeSeedPath : string
+AccountManager()
AccountManagerProvider
accountManager : IBaseAccountAccess
AccountManagerProvider()
+getAccountManagerInstance() : IBaseAccountAccess
Content Management Communication
<<Interface>>
IContentManagement
ContentManager
-categoryManager : ICategoryManager
accesses IBaseAccountAccess interface
Provides singleton instance of AccountManager.
Powered By Visual Paradigm Community Edition

```
IContentManagement
{
    // Note management functions

    createNewNote(note : Note) : int
    searchNoteBy(expression : string) : Note[]
    updateNoteContent(target : Note, newContent : string) : int
    updateNoteCategory(target : Note, newCategory : string) : int

    createNewReminder(reminder : Reminder) : int
    searchReminderBy(expression : string) : Reminder[]
    updateReminderDate(target : Reminder, newDate : Date) : int
    updateReminderInterval(target : Reminder) : int

    sendEmailFor(reminders : Reminder[]) : int

    deleteNote(target : Note) : int

    // Categrory Management functions

    createNewCategory(category : Category) : int
    searchCategoryBy(title : string, owner : string) : Category[]

    updateCategoryTitle(newTitle : string, oldTitle : string, owner : string) : int
    updateCategoryColor(newColor : string, title : string, owner : string) : int
    deleteCategory(title : string, owner : string) : int
}
```


Data Classes
Account Data
AccountInfo
-username : string
-email : string
-settings : string[]
+AccountInfo(username : string, email : string, settings : string[])
+getUsername() : string
+getEmail() : string
+getSettings() : string[]
1
AccountRecord
-info : AccountInfo
-password : string
-sessionID : string = ""
+AccountRecord(info : AccountInfo, password : string)
+getInfo() : AccountInfo
Content Data
Note
-content : string
-owner : string
-category : string
+Note(content : string, owner : string, category : string)
+getContent() : string
+getOwner() : string
+getCategory() : string
Date
-year : int
-month : string
-hour : int
-minute : int
+Date(year : int, month : string, hour : int, minute : int)
+getYear() : int
+getMonth() : string
+getHour() : int
+getMinute() : int
1
Reminder
-date : Date
-interval : int
+Reminder(content : string, owner : string, category : string, date : Date, interval : int)
+getDate() : Date
+getInterval() : int
Category
-name : string
-color : string
-owner : string
+Category(name : string, color : string, owner : string)
+getColor() : string
+getName() : string
+getOwner() : string
Powered By Visual Paradigm Community Edition

UserDB

user
- user_id INT
- username VARCHAR(16)
- password VARCHAR(32)

Indexes

user_info
- user_id INT
- accountInfo VARCHAR(45)
- email VARCHAR(45)

Indexes

user_session
- session_id VARCHAR(45)
- user_id INT

Indexes

user_note
- user_id INT
- note_id INT

Indexes

NoteDB

reminder
- reminder_id INT
- reminder_name VARCHAR(45)
- reminder_date DATETIME
- reminder_interval DATETIME

Indexes

note
- note_id INT
- note_content VARCHAR(45)
- note_create_time DATETIME
- reminder_id INT
- category_id INT

Indexes

note_category
- category_id INT
- category_name VARCHAR(255)
- category_color VARCHAR(45)

Indexes

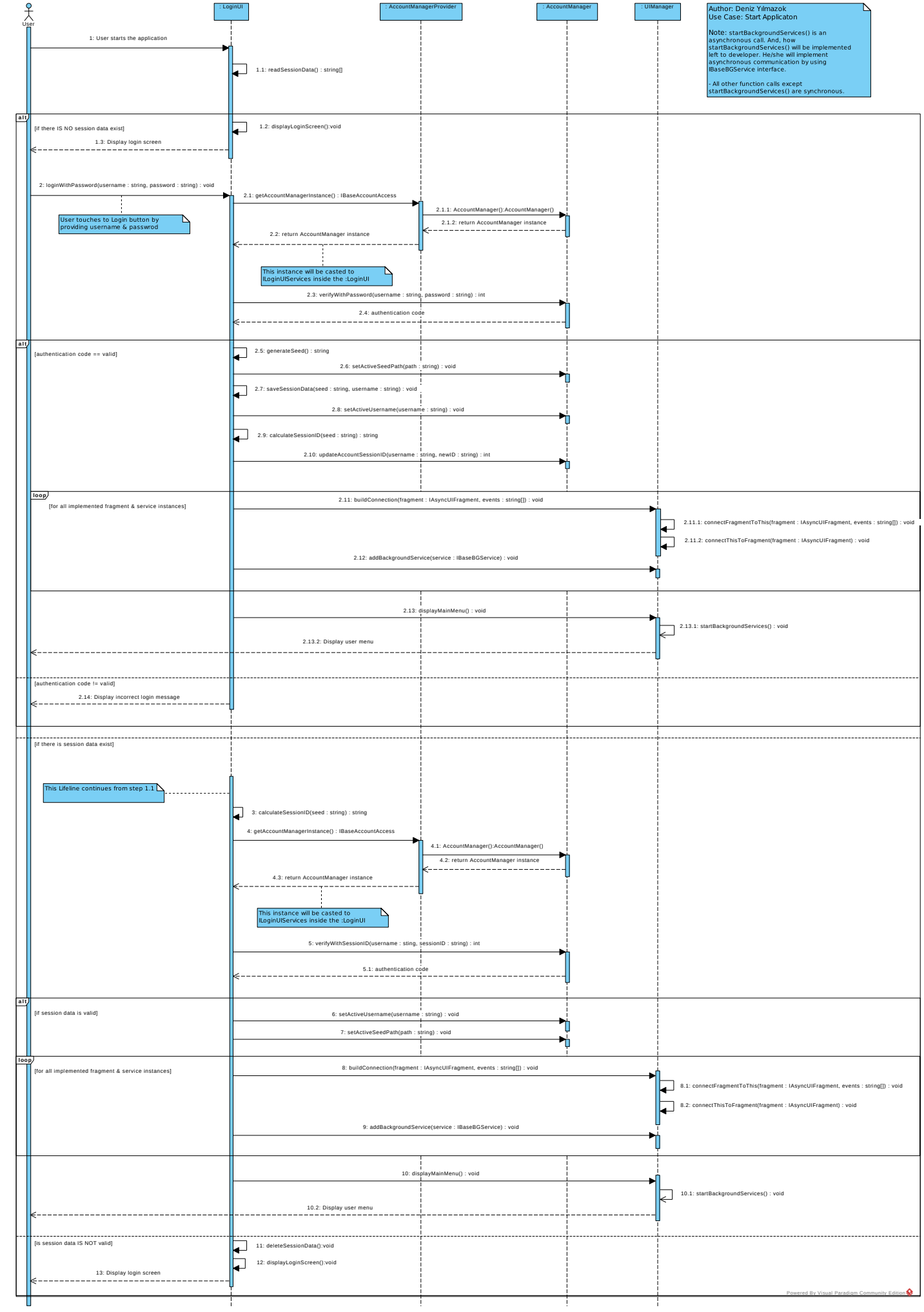
Use Case	LoginUI	UIManager	AccountManagementUI	AccountManager	AccountManagerProvider	ContentManagementUI	ContentManager	NotificationService	ShortcutManagementService
Manage Notes		onNotify()		getActiveUsername()	getAccountManagerInstance()	display() notifyManager() displayNewRecordAddingControls() addNewRecord() downloadNotesBy() downloadCategoriesBy() displayThumbnails() displaySelected(: Note) saveChanges(: Note) deleteSelected(: Note)	createNewNote() searchNoteBy() updateNoteContent() updateNoteCategory() createNewReminder() searchReminderBy() updateReminderDate() updateReminderInterval() deleteNote() createNewCategory() searchCategoryBy() deleteCategory()		
Manage Categories		onNotify()		getActiveUsername()	getAccountManagerInstance()	display() notifyManager() displayNewRecordAddingControls() addNewRecord() downloadCategoriesBy() displayThumbnails() displaySelected(: Category) saveChanges(: Category) deleteSelected(: Category)	createNewCategory() searchCategoryBy() updateCategoryTitle() updateCategoryColor() deleteCategory()		
Manage Fast Notes				getActiveUsername()	getAccountManagerInstance()		createNewNote() searchNoteBy() searchReminderBy()		downloadShortcutContent() displayShortcutWith() displaySelected() displayFastNoteAddingControls() addFastNote()
Manage Account		onNotify()	display() notifyManager() displayAccountInfo() saveAccountChanges() savePassword()	getActiveUsername() getAccountInfo() updateAccountPassword() updateAccountEmail() updateAccountSettings()	getAccountManagerInstance()				

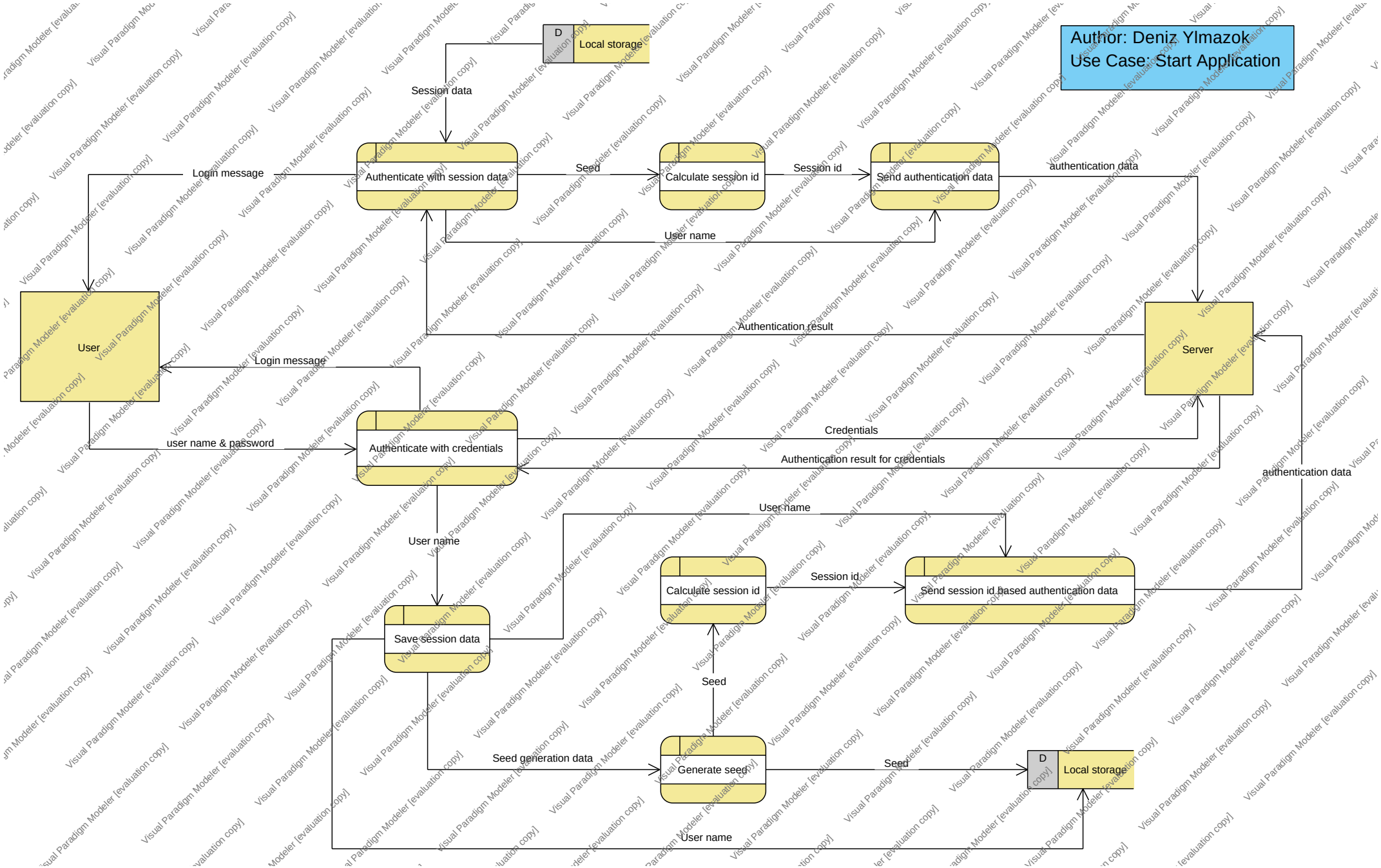
Close Application		onNotify() stopBackgroundServices()	display() notifyManager() exitFromApplication() deleteActiveSeed()	getActiveUsername() cancelSessionID() getActiveSeedPath()	getAccountManagerInstance()			stop() onStopTriggered()	stop() onStopTriggered()
Start Application	displayLoginScreen() loginWithPassword() generateSeed() saveSessionData() readSessionData() calculateSessionID() loginWithSessionID() deleteSessionData()	connectFragmentToThis() connectThisToFragment() buildConnection() displayMainMenu() addBackgroundService() startBackgroundServices()		verifyWithPassword() verifyWithSessionID() updateAccountSessionID() setActiveUsername() setActiveSeedPath()	getAccountManagerInstance()			start() onStartTriggered() isAlreadyRunning()	start() onStartTriggered() isAlreadyRunning()
Create New Account	displayAccountCreationScreen() createAccount() displayLoginScreen()			createNewAccount()	getAccountManagerInstance()				
Send Notification to User for Reminder(s)				getActiveUsername() getAccountInfo()	getAccountManagerInstance()		updateReminderDate()	isReminderExist() downloadNotificationContent() downloadNotificationSettings() notify() sendEmail() displayPostponeOptions() postpone()	

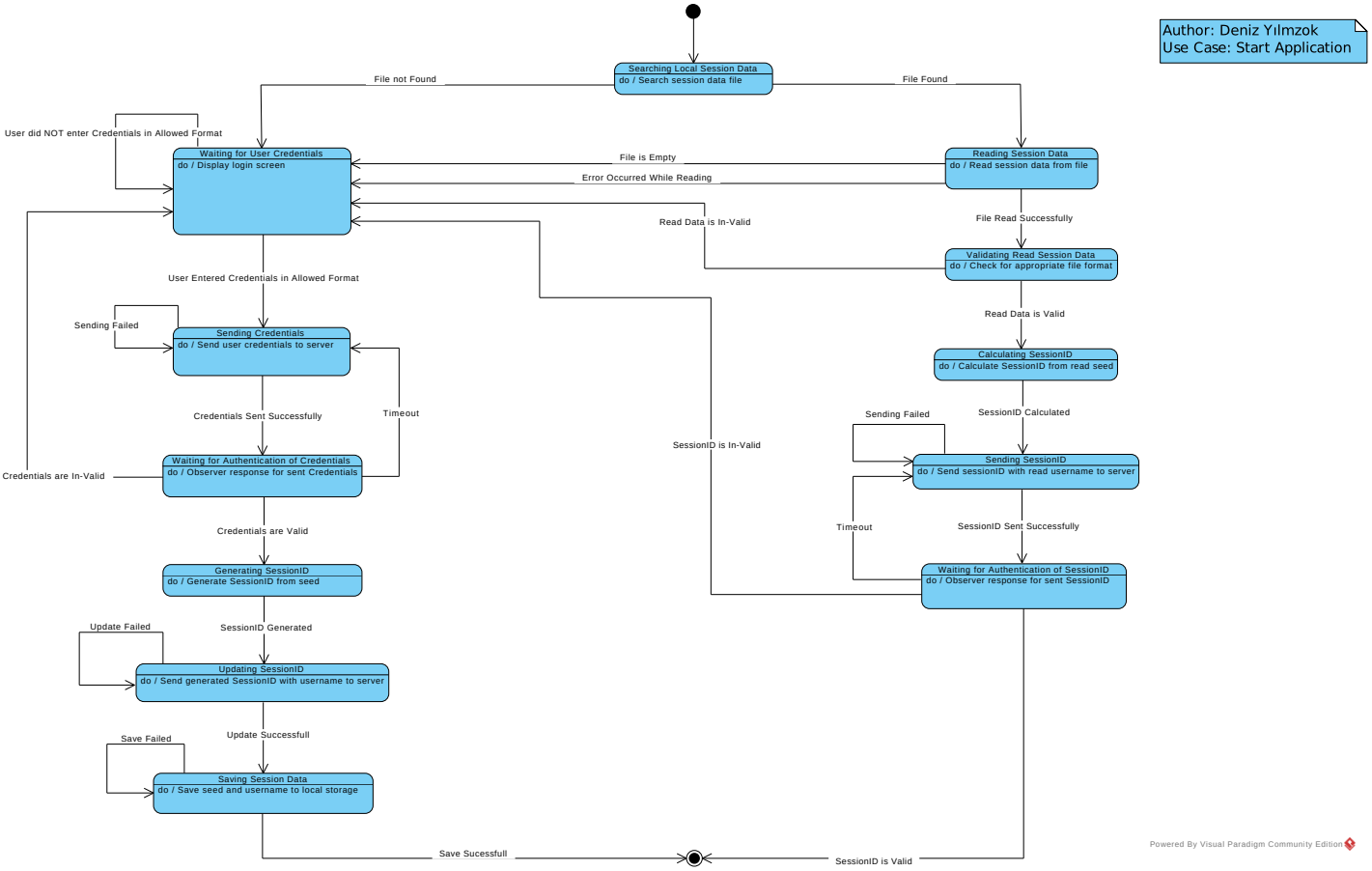
Notes:
All functions' return type and arguments are excluded for saving space. To see them in detail, please check the class diagram and interface definitions section.
Overridden functions given with their arguments.

UIManager.hide() and UImanager.receiveEvent() are not included in this chart because they will be triggered by unexpected behaviours of the user. That is why they are rarely executed. Also, developer will benefit from these functions by implementing them according to his/her needs.

View Tasks from Menu is merged with Manage Notes.
Add Fast Note from Shortcut and View Tasks from Shortcut are merged to Manage Fast Notes.


User
: LoginUI
: AccountManagerProvider
: AccountManager
: UIManager
Author: Deniz Yilmazok
Use Case: Start Applicaton
Note: startBackgroundServices() is an asynchronous call. And, how startBackgroundServices() will be implemented left to developer. He/she will implement asynchronous communication by using IBaseBGService interface.
- All other function calls except startBackgroundServices() are synchronous.
1: User starts the application
1.1: readSessionData() : string[]
alt
[if there IS NO session data exist]
1.2: displayLoginScreen():void
1.3: Display login screen
2: loginWithPassword(username : string, password : string) : void
User touches to Login button by providing username & passwrod
2.1: getAccountManagerInstance() : IBaseAccountAccess
2.1.1: AccountManager():AccountManager()
2.1.2: return AccountManager instance
2.2: return AccountManager instance
This instance will be casted to ILoginUIServices inside the :LoginUI
2.3: verifyWithPassword(username : string, password : string) : int
2.4: authentication code
alt
[authentication code == valid]
2.5: generateSeed() : string
2.6: setActiveSeedPath(path : string) : void
2.7: saveSessionData(seed : string, username : string) : void
2.8: setActiveUsername(username : string) : void
2.9: calculateSessionID(seed : string) : string
2.10: updateAccountSessionID(username : string, newID : string) : int
loop
[for all implemented fragment & service instances]
2.11: buildConnection(fragment : IAsyncUIFragment, events : string[]) : void
2.11.1: connectFragmentToThis(fragment : IAsyncUIFragment, events : string[]) : void
2.11.2: connectThisToFragment(fragment : IAsyncUIFragment) : void
2.12: addBackgroundService(service : IBaseBGService) : void
2.13: displayMainMenu() : void
2.13.1: startBackgroundServices() : void
2.13.2: Display user menu
[authentication code != valid]
2.14: Display incorrect login message
[if there is session data exist]
This Lifeline continues from step 1.1
3: calculateSessionID(seed : string) : string
4: getAccountManagerInstance() : IBaseAccountAccess
4.1: AccountManager():AccountManager()
4.2: return AccountManager instance
4.3: return AccountManager instance
This instance will be casted to ILoginUIServices inside the :LoginUI
5: verifyWithSessionID(username : sting, sessionID : string) : int
5.1: authentication code
alt
[if session data is valid]
6: setActiveUsername(username : string) : void
7: setActiveSeedPath(path : string) : void
loop
[for all implemented fragment & service instances]
8: buildConnection(fragment : IAsyncUIFragment, events : string[]) : void
8.1: connectFragmentToThis(fragment : IAsyncUIFragment, events : string[]) : void
8.2: connectThisToFragment(fragment : IAsyncUIFragment) : void
9: addBackgroundService(service : IBaseBGService) : void
10: displayMainMenu() : void
10.1: startBackgroundServices() : void
10.2: Display user menu
[is session data IS NOT valid]
11: deleteSessionData():void
12: displayLoginScreen():void
13: Display login screen
Powered By Visual Paradigm Community Edition


Author: Deniz Ylmazok
Use Case: Start Application
D
Local storage
Session data
Login message
Authenticate with session data
Seed
Calculate session id
Session id
Send authentication data
authentication data
User name
User
Authentication result
Server
Login message
user name & password
Authenticate with credentials
Credentials
Authentication result for credentials
authentication data
User name
User name
Calculate session id
Session id
Send session id based authentication data
Save session data
Seed
Seed generation data
Generate seed
Seed
D
Local storage
User name


Author: Deniz Yılmzok
Use Case: Start Application
Searching Local Session Data
do / Search session data file
File not Found
File Found
User did NOT enter Credentials in Allowed Format
Waiting for User Credentials
do / Display login screen
File is Empty
Error Occurred While Reading
Reading Session Data
do / Read session data from file
Read Data is In-Valid
File Read Successfully
Validating Read Session Data
do / Check for appropriate file format
User Entered Credentials in Allowed Format
Read Data is Valid
Sending Failed
Sending Credentials
do / Send user credentials to server
Calculating SessionID
do / Calculate SessionID from read seed
Credentials Sent Successfully
Timeout
SessionID Calculated
Sending Failed
SessionID is In-Valid
Credentials are In-Valid
Waiting for Authentication of Credentials
do / Observer response for sent Credentials
Sending SessionID
do / Send sessionID with read username to server
Timeout
SessionID Sent Successfully
Credentials are Valid
Generating SessionID
do / Generate SessionID from seed
Waiting for Authentication of SessionID
do / Observer response for sent SessionID
Update Failed
SessionID Generated
Updating SessionID
do / Send generated SessionID with username to server
Save Failed
Update Successfull
Saving Session Data
do / Save seed and username to local storage
Save Sucessfull
SessionID is Valid
Powered By Visual Paradigm Community Edition