Architectural and design

Group: Shahd Mustafa, Mike Nasser, Nathaniel Leonardo, Hassan Radwan, Hussen Al-Jubury

Introduction

- Purpose
  - Overview of the project, taking a look at the main goals, objectives, and expected challenges

- Scope
  - A brief look at the intended scope of the splitsmart application

Architectural Goals

1. Scalability
2. Reliability
3. Security
4. Usability

Architecture Restraints

1. Platform compatibility
2. Performance
3. Integration
4. Compliance

Use case view

- **User login**
- User registration
- **Payment information**
- Manage groups
- **New expense**
- Split expense
- Assign expense
- **Customer report**
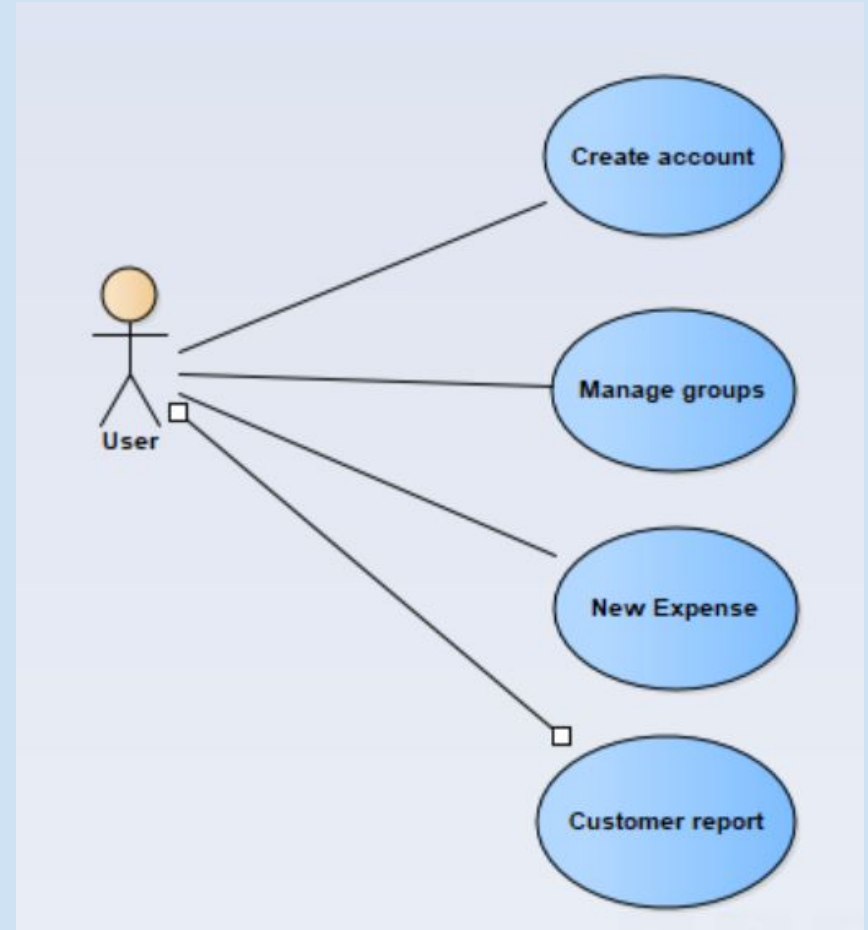- Expense notification
- **Payment**
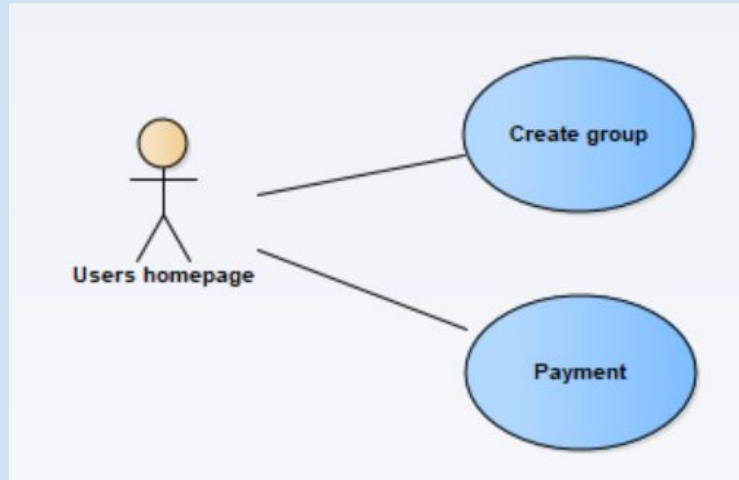- **Homepage**

Figure 1: user account use case
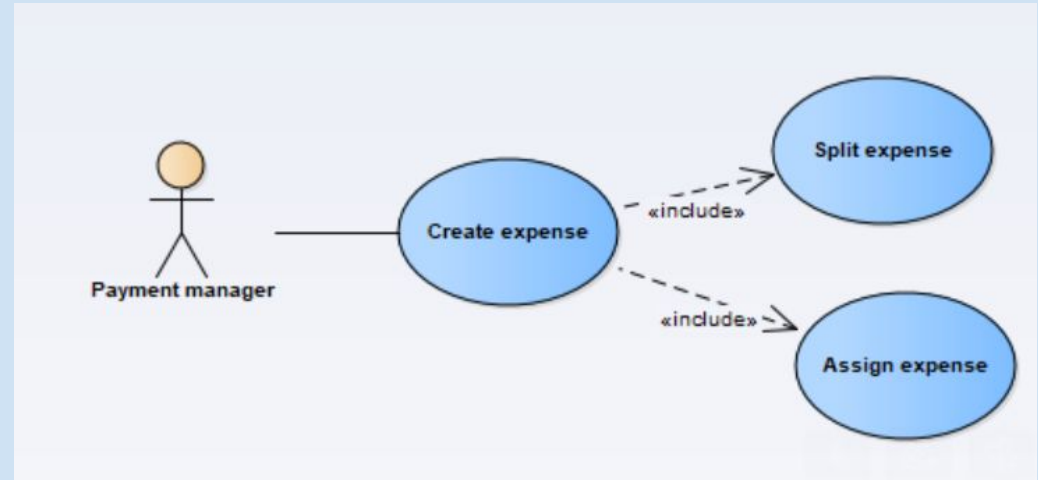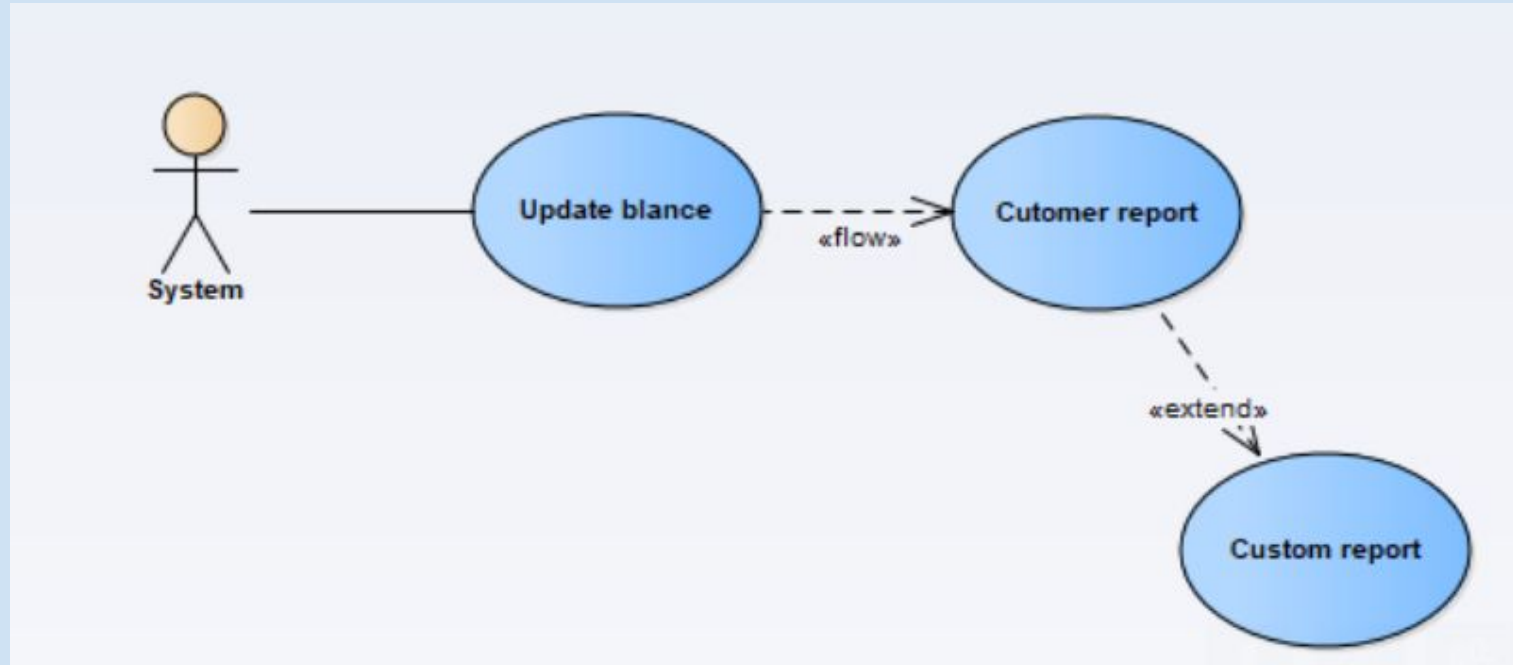
Figure 2: creating expense use case

Figure 3: balance update use case
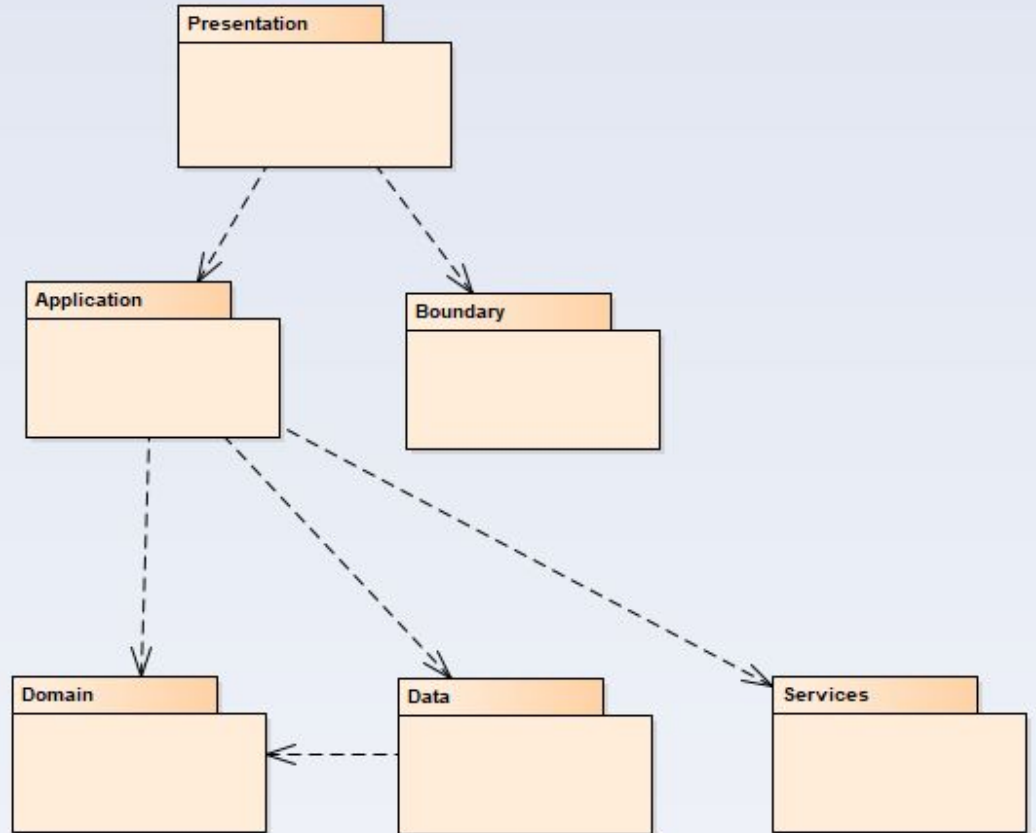
Significant use case descriptions

- User login
- User registration
- Payment information
- Manage groups
- New expense
- Split expense
- Assign expense

- Customer report
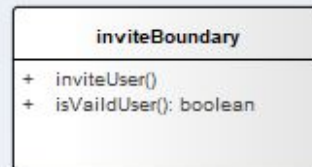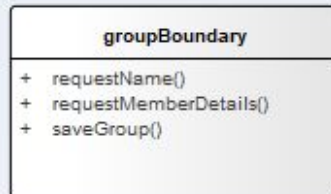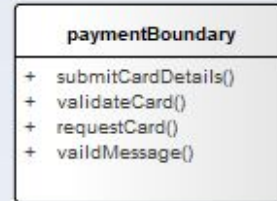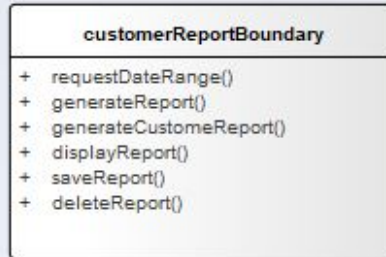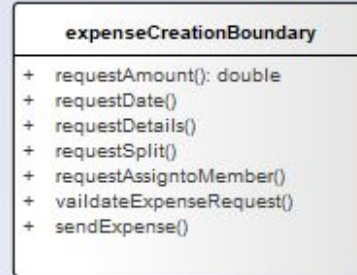- Expense notification
- Payment
- Homepage

Logical view

Packages:
- Presentation
- Boundary
- Application
- Domain
- Data
- Services

Logical view

accountBoundary

+ editeAccount()
+ editPayemnt()
+ displayProfileInfo()
+ submitEdits()
+ selectCatagory()

expenseCreationBoundary

+ requestAmount(): double
+ requestDate()
+ requestDetails()
+ requestSplit()
+ requestAssigntoMember()
+ vaildateExpenseRequest()
+ sendExpense()

customerReportBoundary

+ requestDateRange()
+ generateReport()
+ generateCustomeReport()
+ displayReport()
+ saveReport()
+ deleteReport()

paymentBoundary

+ submitCardDetails()
+ validateCard()
+ requestCard()
+ vaildMessage()

groupBoundary

+ requestName()
+ requestMemberDetails()
+ saveGroup()

inviteBoundary

+ inviteUser()
+ isVaildUser(): boolean

Local view

Presentation package

accountControl
+ getProfileInformation()
+ validateProfileInfo()
+ testforNewAccounts()

expenseCreationControl
+ getExpense()
+ updateBalance()
+ submitExpense()

paymentControl
+ updateCardInfo()

groupControl
+ updateGroup()

notificationControl
+ submitNotification()

customerReportControl
+ updateReports()
+ generateReport(): int

inviteControl
+ getMember()
+ addMember()

Application package

UserPage

Support

Profile

Account

Logical view

Domain package

User page Package

userProfile
- userName
- password
- cardInformation

ProfileContent

GroupProfile
+ updateGroups()
+ getGroupsIN()
+ getBalance()
+ updateBalance()
+ getPayment()
+ getName()
+ getMembers()
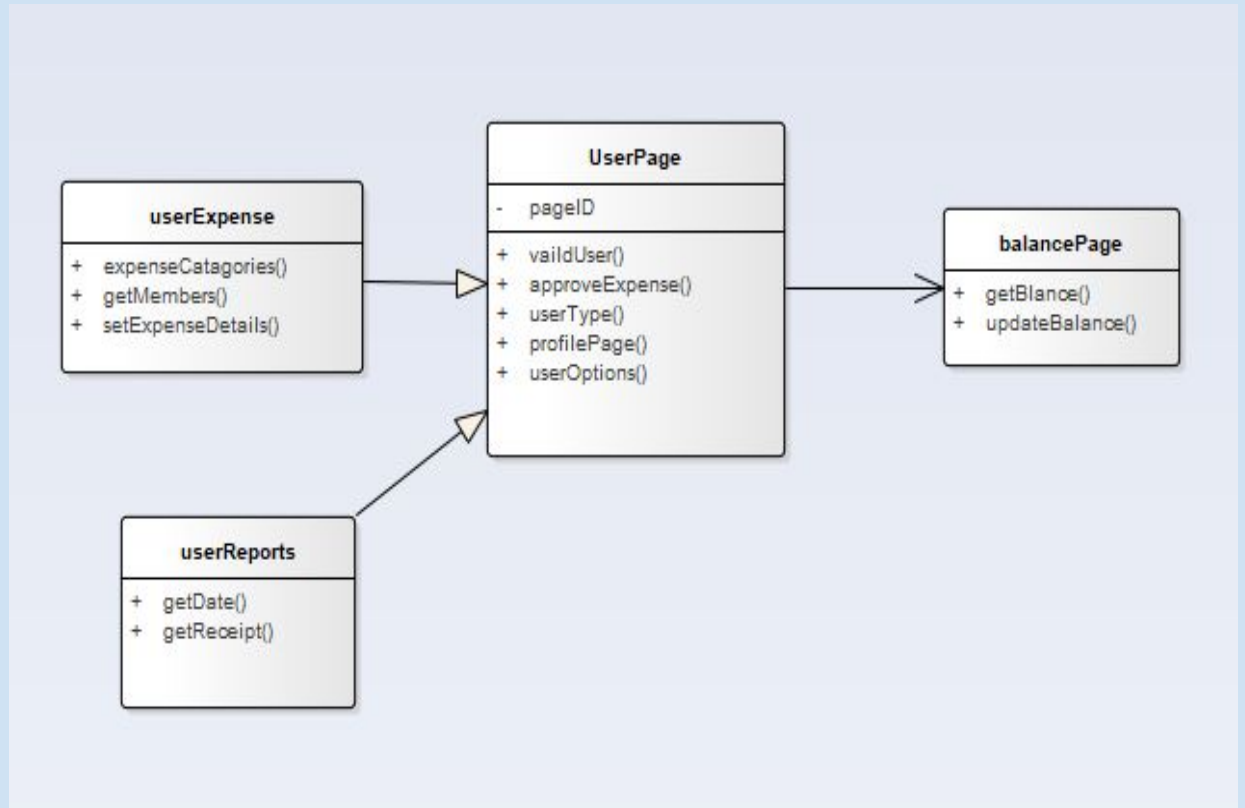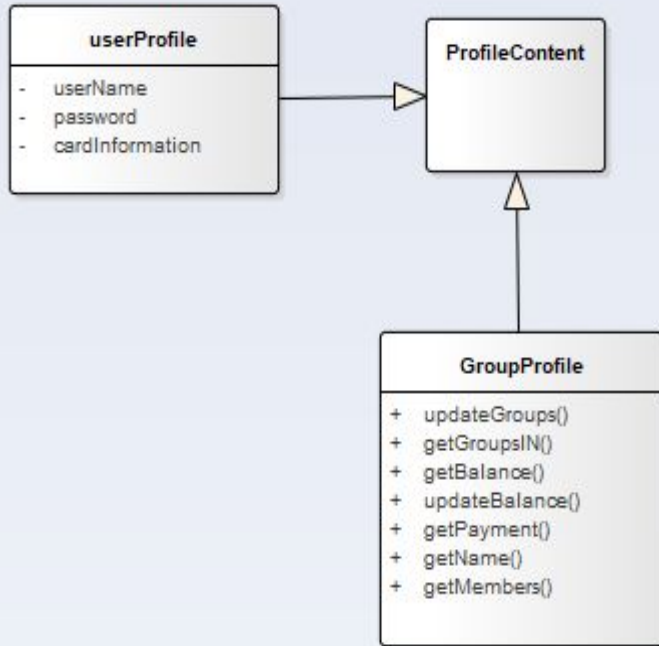
Profile package

AccountTermsandCondition

userProfile
- userName
- password
- cardInformation

ProfileContent

GroupProfile
+ updateGroups()
+ getGroupsIN()
+ getBalance()
+ updateBalance()
+ getPayment()
+ getName()
+ getMembers()
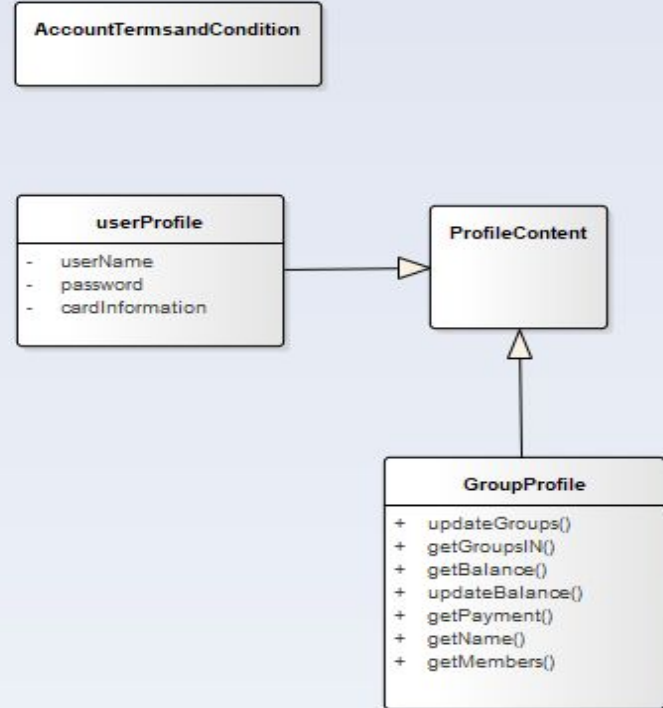
Account package

SupportDetails

- supportEmailDetails
- supportPhoneDetails

+ requestSupportNumber(): int
+ requestSupportEmail()
+ sendManual()

Support package

Database

+ UpdateDB()
+ deleteDB()
+ addDB()
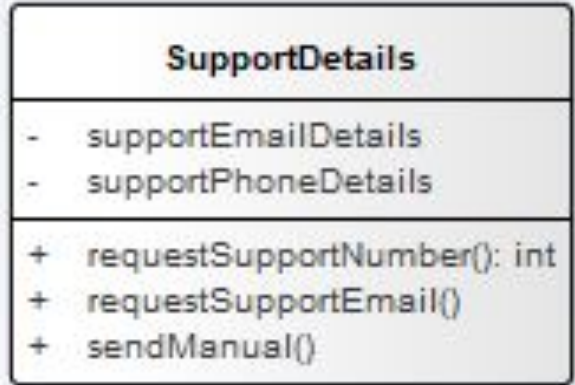
ProfileDatabase
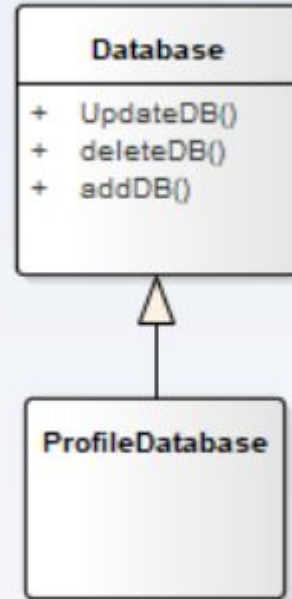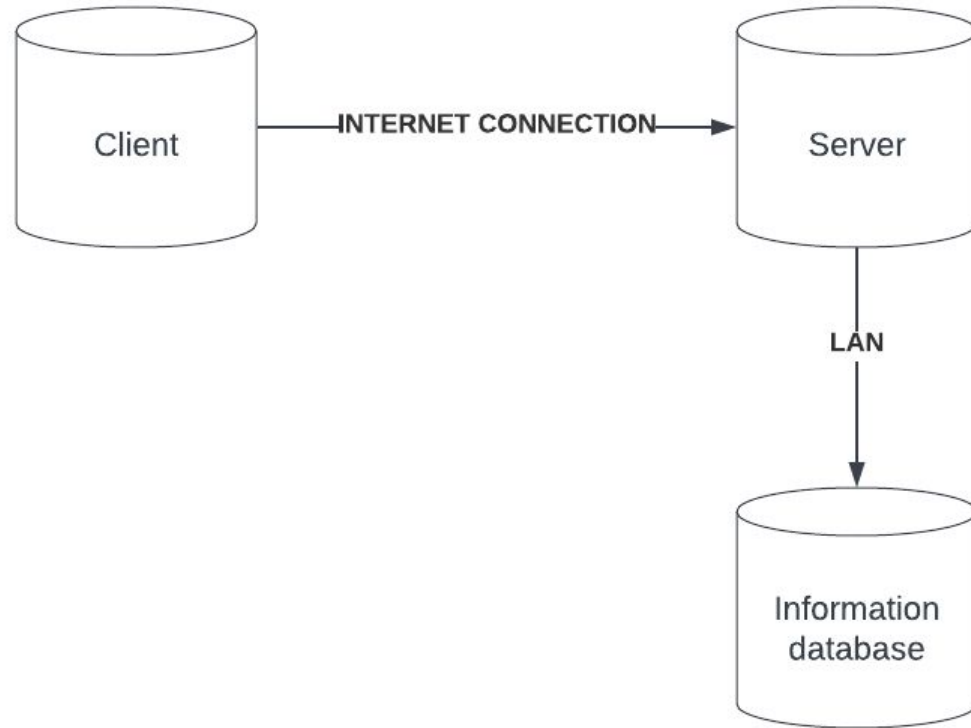
Deployment view

Size and Performance

Roughly 10,000 Online users simultaneously

Any number of users using application locally

Quality

- User Experience
- Minimize Downtime
- Well Designed Architecture
- Security
- Extensibility
- Testability