UML

CASH MANAGER

Team: 7

Lucas Boisbourdin

Bryan Lebar

Barthelmebs Alexis

Julien Barriere

Packages Arrangement

Startup

+ «property» Configuration: IConfiguration

+ «create» Startup (configuration : IConfiguration)
+ ConfigureServices (services : IServiceCollection)
+ Configure (app : IApplicationBuilder, env : IWebHost Environment)

Program

+ Main (args : string[])
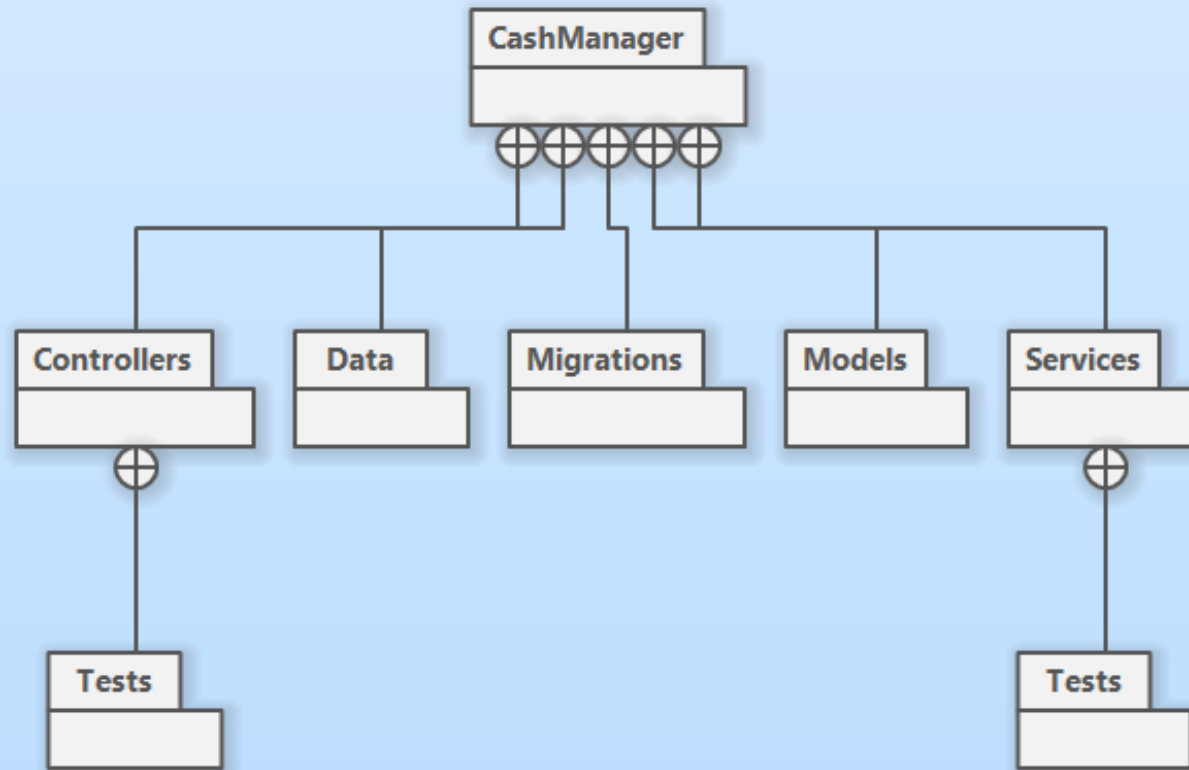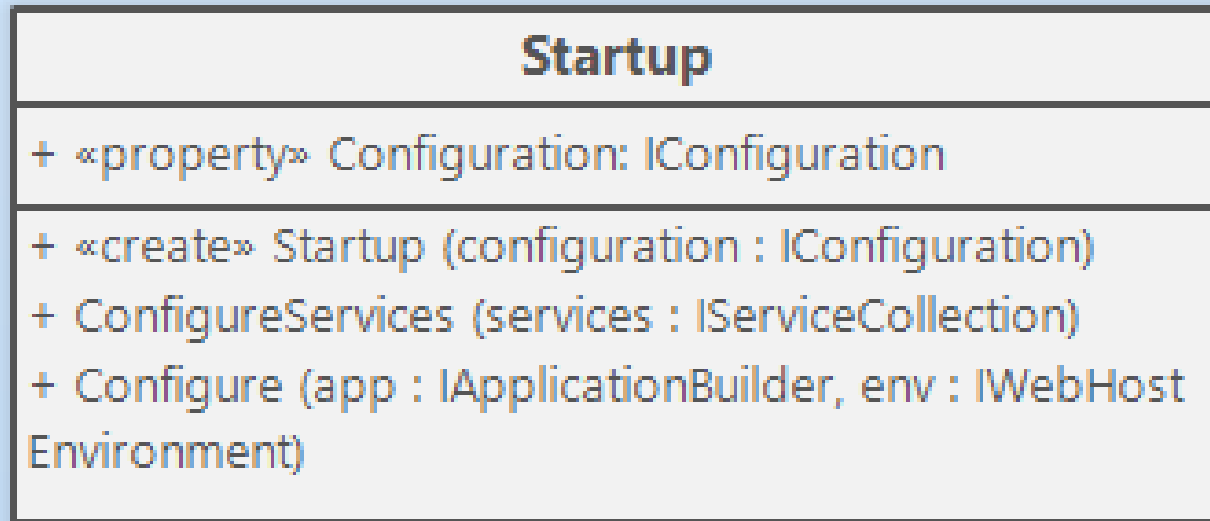
Classes - CashManager

UserController

- _userService: IUserService {readOnly}
- _productService: IProductService {readOnly}

+ «create» UserController (userService : IUserService, productService : IProductService)
+ Index (): JsonResult
+ Login (username : string, password : string): JsonResult
+ GetProductPrice (productReference : string): JsonResult
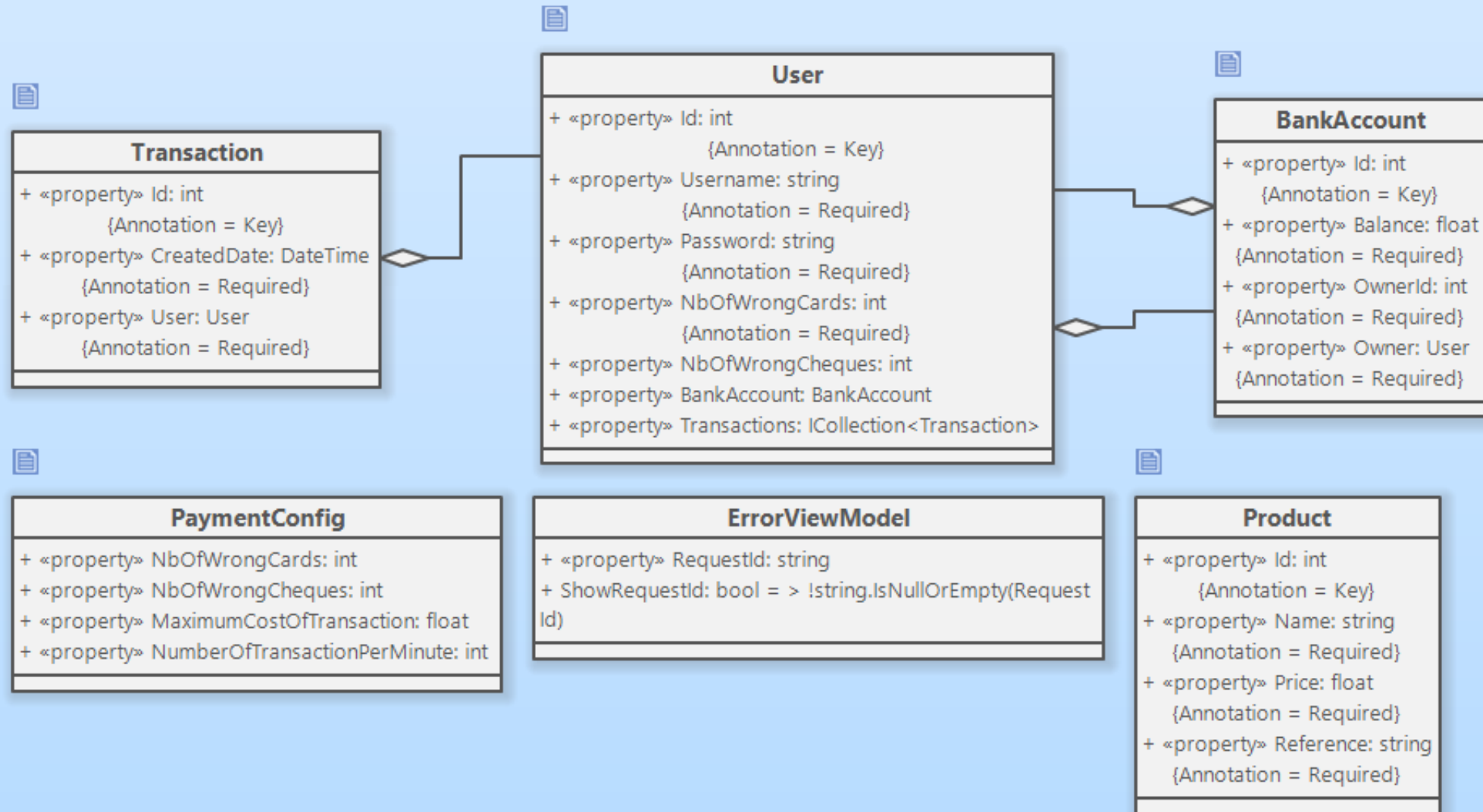+ Pay (userId : int, ammount : float, isCreditCard : bool): JsonResult

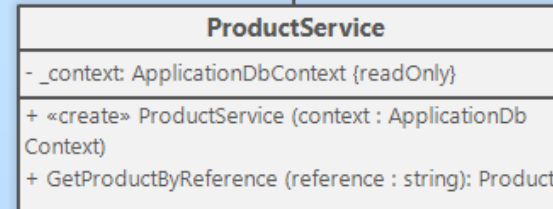{Annotation = HttpPost}

Classes – Controller - UserController
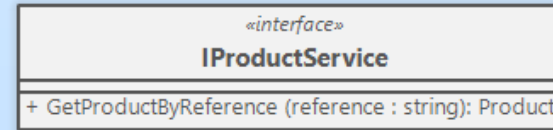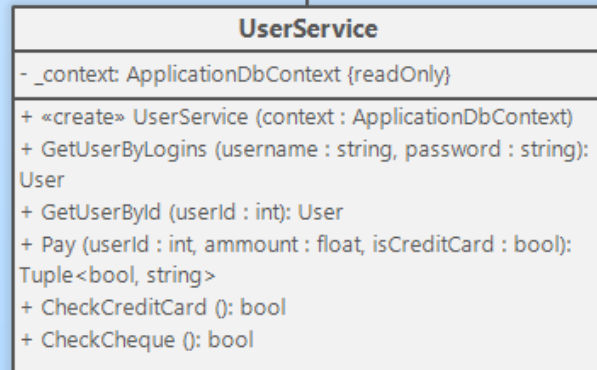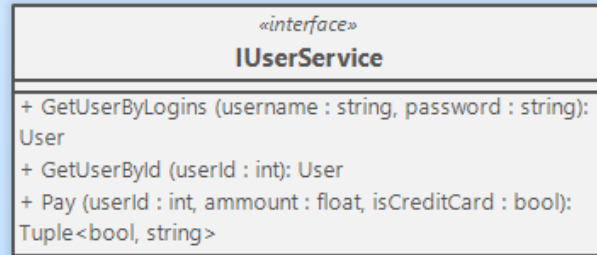
ApplicationDbContext

+ «property» Users: DbSet<User>

+ «property» Transactions: DbSet<Transaction>

+ «property» BankAccounts: DbSet<BankAccount>

+ «property» Products: DbSet<Product>

+ «create» ApplicationDbContext (options : DbContext Options<ApplicationDbContext>)

OnModelCreating (modelBuilder : ModelBuilder)

Classes – Data - ApplicationDbContext

Classes - Models

Transaction

+ «property» Id: int
    {Annotation = Key}
+ «property» CreatedDate: DateTime
    {Annotation = Required}
+ «property» User: User
    {Annotation = Required}

User

+ «property» Id: int
    {Annotation = Key}
+ «property» Username: string
    {Annotation = Required}
+ «property» Password: string
    {Annotation = Required}
+ «property» NbOfWrongCards: int
    {Annotation = Required}
+ «property» NbOfWrongCheques: int
+ «property» BankAccount: BankAccount
+ «property» Transactions: ICollection<Transaction>

BankAccount

+ «property» Id: int
    {Annotation = Key}
+ «property» Balance: float
    {Annotation = Required}
+ «property» OwnerId: int
    {Annotation = Required}
+ «property» Owner: User
    {Annotation = Required}

PaymentConfig

+ «property» NbOfWrongCards: int
+ «property» NbOfWrongCheques: int
+ «property» MaximumCostOfTransaction: float
+ «property» NumberOfTransactionPerMinute: int

ErrorViewModel

+ «property» RequestId: string
+ ShowRequestId: bool = > !string.IsNullOrEmpty(RequestId)

Product

+ «property» Id: int
    {Annotation = Key}
+ «property» Name: string
    {Annotation = Required}
+ «property» Price: float
    {Annotation = Required}
+ «property» Reference: string
    {Annotation = Required}

Classes - Service

«interface»
IUserService

+ GetUserByLogins (username : string, password : string):
User
+ GetUserById (userId : int): User
+ Pay (userId : int, ammount : float, isCreditCard : bool):
Tuple<bool, string>

UserService

- _context: ApplicationDbContext {readOnly}

+ «create» UserService (context : ApplicationDbContext)
+ GetUserByLogins (username : string, password : string):
User
+ GetUserById (userId : int): User
+ Pay (userId : int, ammount : float, isCreditCard : bool):
Tuple<bool, string>
+ CheckCreditCard (): bool
+ CheckCheque (): bool

«interface»
IProductService

+ GetProductByReference (reference : string): Product

ProductService

- _context: ApplicationDbContext {readOnly}

+ «create» ProductService (context : ApplicationDb
Context)
+ GetProductByReference (reference : string): Product

Classes – Test – UserController Test

UserServiceTest

+ GetUserByLoginsTest ()
+ GetUserByLoginsTestFail ()
  {Annotation = TestMethod()}
+ GetUserByIdTestSuccess ()
+ GetUserByIdTestFail ()
+ PayTestCreditCardSucess ()
+ PayTestChequeSucess ()
+ PayTestChequeFailAttempt ()
+ PayTestChequeFailFund ()
+ PayTestCreditCardFailAttempt ()
+ PayTestCreditCardFailFund ()
  {Annotation = TestMethod}
+ PayTestFailTooManyTransactions ()
  {Annotation = TestMethod()}
+ CheckCreditCardTest ()
  {Annotation = TestMethod()}
+ CheckChequeTest ()
  {Annotation = TestMethod()}

Classe – Test-UserService Test
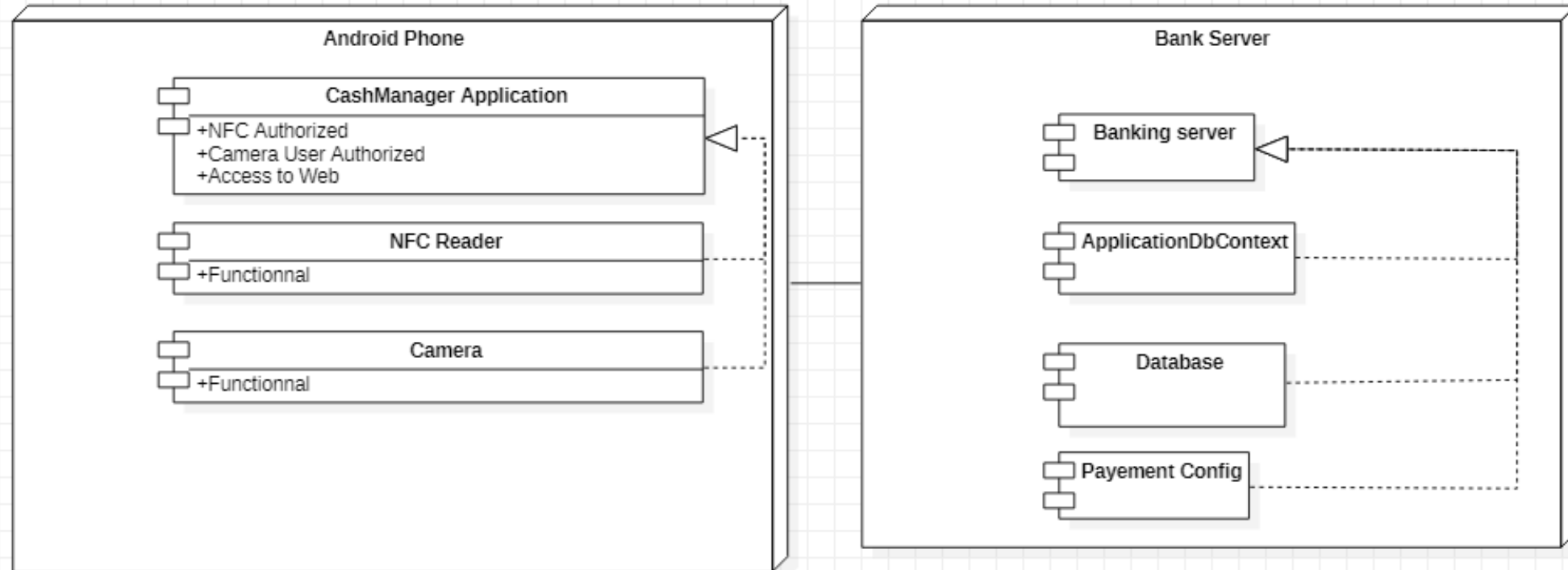
Database schéma

Migration History

Use Case Diagram - App

Sequence Diagram – Best Case Scenario
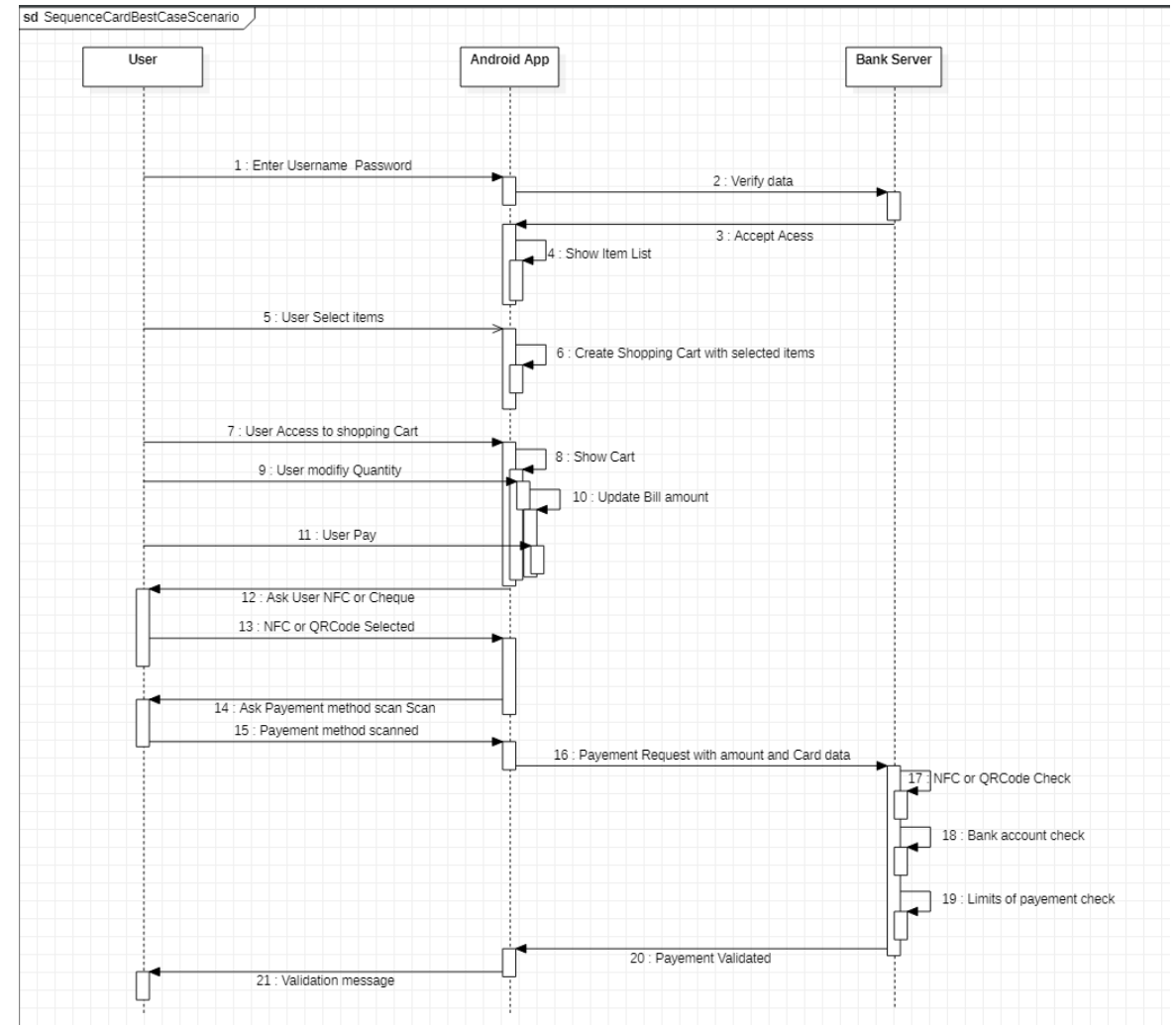
Deployement Diagram

Activity Diagram

Trouvez le PDF ActivityDiag dans le fichier source