

# APSP

## *System Design*

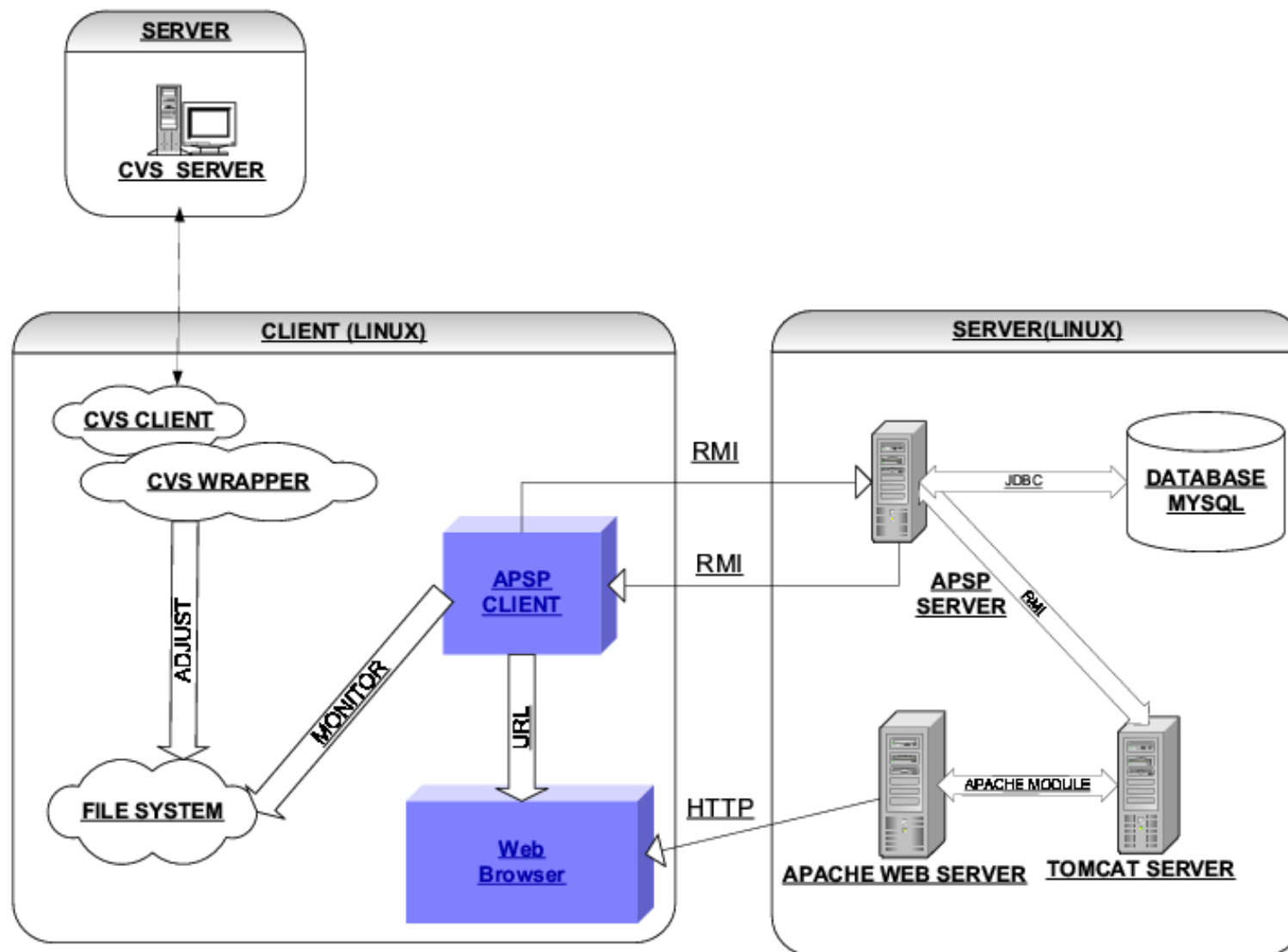
Abraham James, Christopher Rebstock, Kareemullah Quadir,  
Roshan Rammohan, Vinod Doshi, Yaroslav Halchenko

<http://apsp.onerussian.com>

# Introduction

- ☞ general architectural design
- ☞ database schema
- ☞ OO design (classes)
- ☞ example of sequence diagram on specific use case
- ☞ example of data flow diagrams on specific use case.

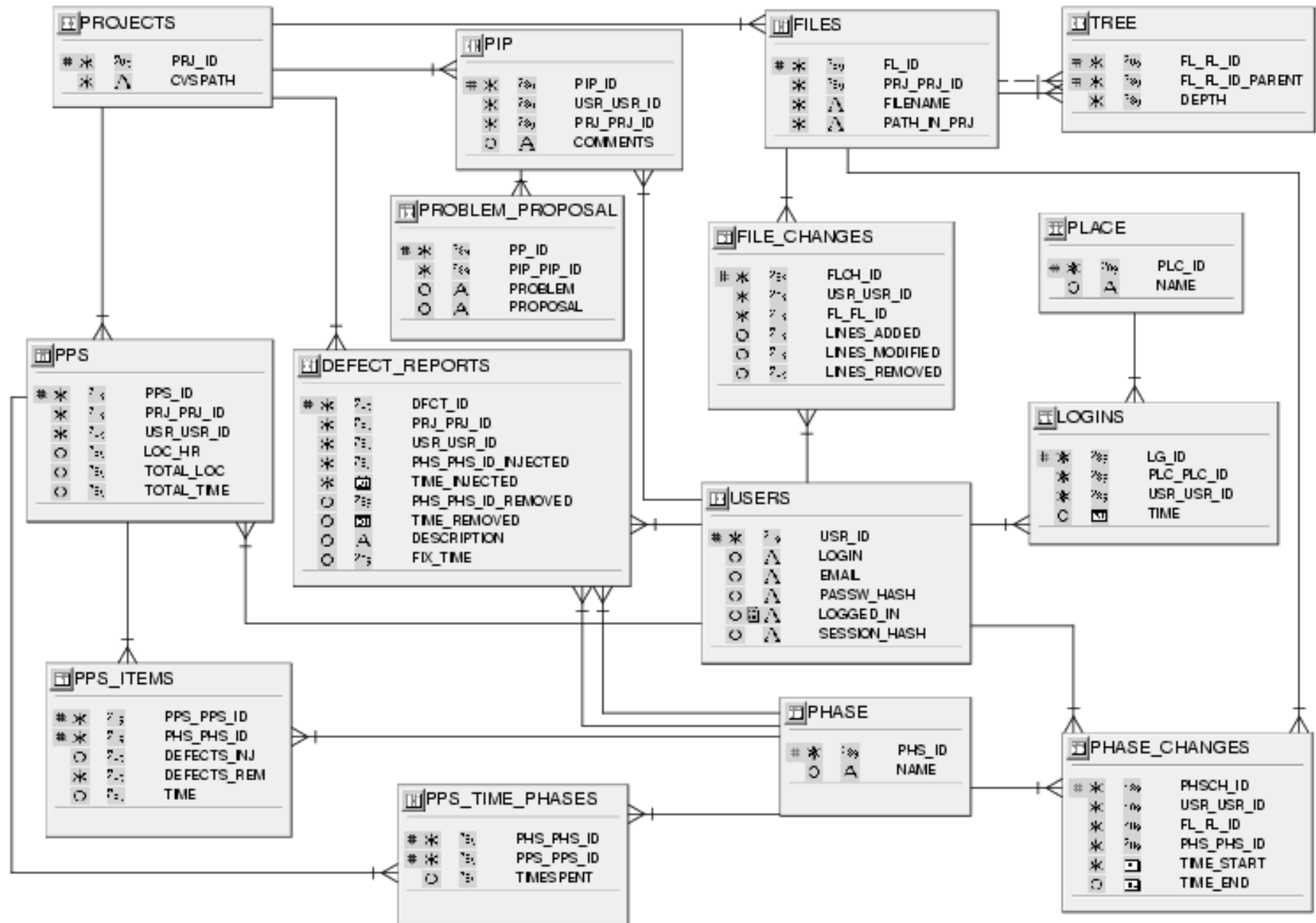
# Architectural Design



# Database Design. TABLES

- USERS
- LOGINS
- PLACE
- PHASE
- PHASE\_CHANGES
- FILES
- TREE
- FILE\_CHANGES
- PROJECTS
- PIP
- PROBLEM\_PROPOSAL
- DEFECT\_REPORTS
- PPS
- PPS\_ITEMS

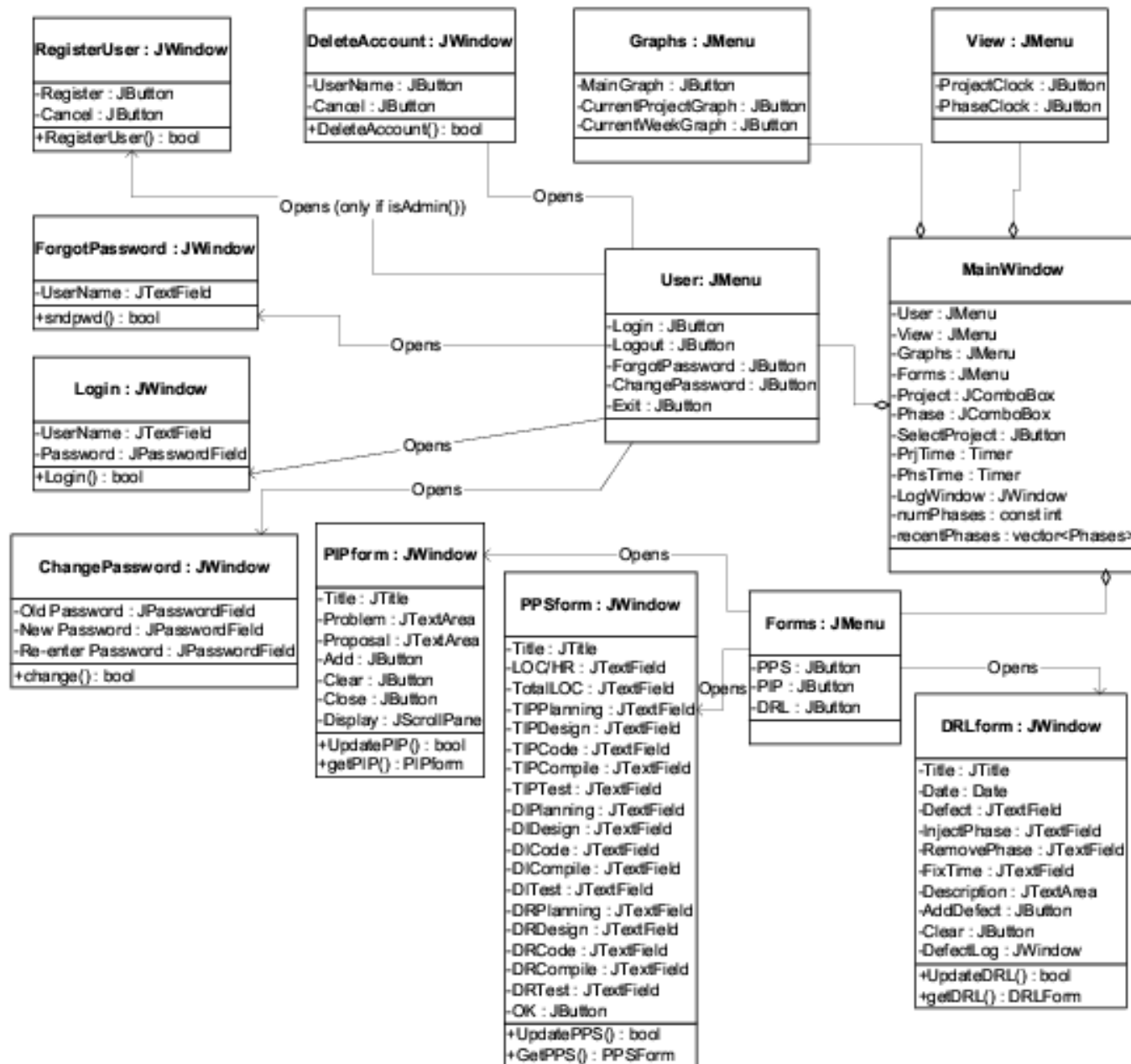
# Database Design. Continued



# Obj. Oriented Design

- ✌ Java User Interface
- ✌ Client-side
- ✌ Server RMI
- ✌ Server-side
- ✌ Serializable

# Obj. Oriented Design. Java UI



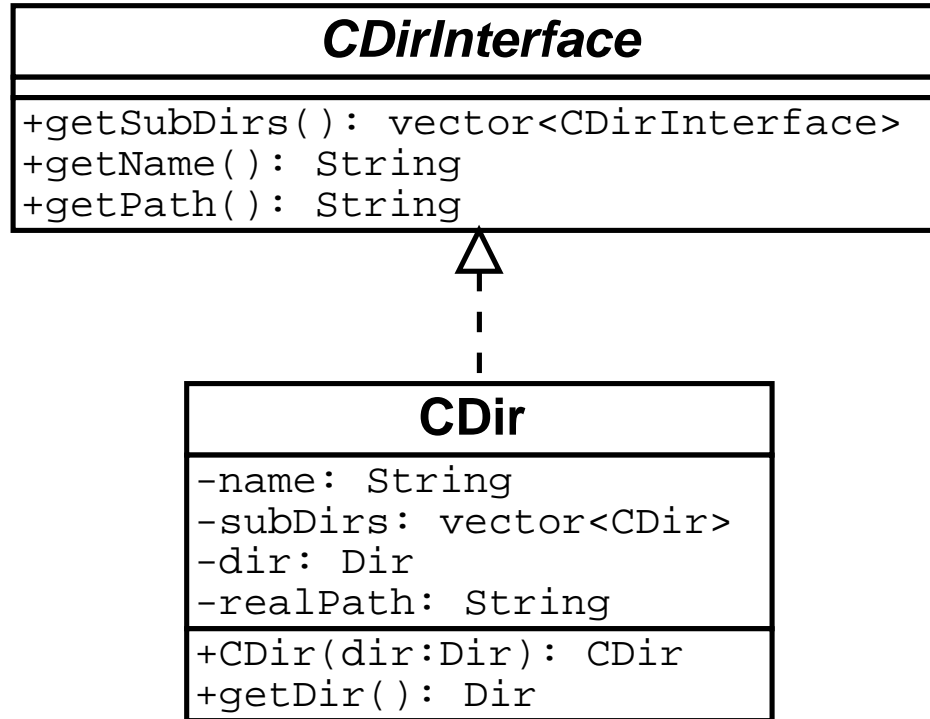
# Class Client (Client-side)

## Client

```
-user: User
-projects: Vector<Project>
-curDir: File
-server: Server
-phases: vector<Phase>

-timer()
-diffDirectory(dir:Dir,dirCur:String,dirOld:String): bool
-checkDirectories(): void
+authenticateUser(userName:String,password:String): String
+deleteUser(userName:String): String
+registerUser(userName:String,email:String): String
+isLoggedIn(): bool
+isAdmin(): bool
+changePassword(oldPassword:String,newPassword:String): String
+forgotPassword(): void
+changeDir(dir:CDir): void
+changePhase(phase:Phase): void
+getMainGraphURL(): String
+getProjGraphURL(): String
+getWeekGraphURL(): String
+getPhases(): vector<Phases>
+getDirs(): vector<CDir>
```

# Class CDir (Client-side)



# Class Server (Server-side)

## Server

-db: DB

-openedProjects: container<Project>

+authenticateUser(login:string,password:string,place:String): User

+getUser(session\_hash:string): User

+getPhases(): vector<Phases>

+getProject(cvspath:String): project

# Class File (Server RMI)

## File

```
-flId: int  
-fileName: String  
-parentProject: Project  
-db: DB
```

```
+File(id:int,name:String,parent:Project,lDb:DB): File  
+register(user:User,added:int,removed:int,modified:int): void  
+changePhase(usr:User,phase:Phase): void  
+getProject(): Project  
+getStat(graphId:int,start:Time,end:Time,usr:User): int  
+getName(): String
```

# Class Dir (Server RMI)

## Dir

+files: vector<File>

+getFiles(): vector<File>

+getStat(graphId:int, start:Time, end:Time, usr:User): int

+createFile(fileName:String): File

# Class Project (Server RMI)

## Project

-prj\_Id: int

+getId(): int

+getRoot(): Dir

+getDefects(usr:User): vector<Defects>

+updateDefects(defects:vector<Defect>,user:User): bool

+updatePIP(pip:PIP): bool

+getPIP(usr:User): PIP

+updatePPS(ppsform:PPS): bool

+getPPS(usr:User): PPS

# Class DB (Server-side)

## DB

-connection: Connection

+openConnection(dbName:string): bool

+isOpen(): bool

+endConnection(): bool

+runQuery(str:string): resultSet

# Class Phase (Serializable)

## Phase

-phaseId: int

-phaseName: String

+Phase(id:int,name:String): void

+getPhaseId(): int

+getPhaseName(): String

# Class PPS (Serializable)

## PPS

```
-totalEstLoc: int
-EstLocPerHour: int
-time: vector<array[3]:int>
-defectsInjected: vector<array[3]:int>
-defectsRemoved: vector<array[3]:int>

+getTotalLoc(field:int): int
+setTotalEstLoc(value:int): void
+getLocPerHour(field:int): int
+setEstLocPerHour(value:int): void
+getField(block:int,phs:Phase,field:int): int
+setField(block:int,phs:Phase,value:int): void
```

# Class PP (Serializable)

## PP

```
-ppId: int  
-problemDescription: String  
-proposalDescription: String  
-ppstatus: int
```

```
+getProblemDescription(): String  
+setProblemDescription(probdesc:String): bool  
+getProposalDescription(): String  
+setProposalDescription(propdesc:String): bool  
+getppStatus(): bool  
+setppStatus(stat:bool): bool
```

# Class PIP (Serializable)

## PIP

```
-pipId: int  
-pipComments: String  
-pipProblemProposals: vector<PP>  
+getComments(): String  
+setComments(comments:String): bool
```

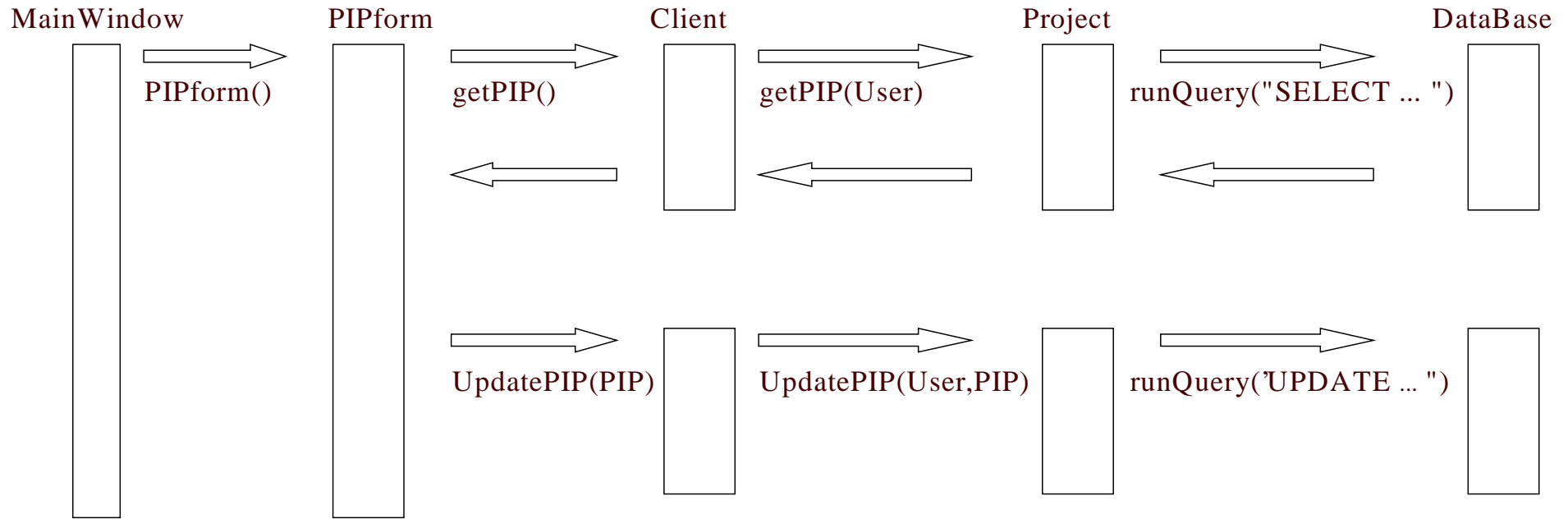
# Class Defect (Serializable)

## Defect

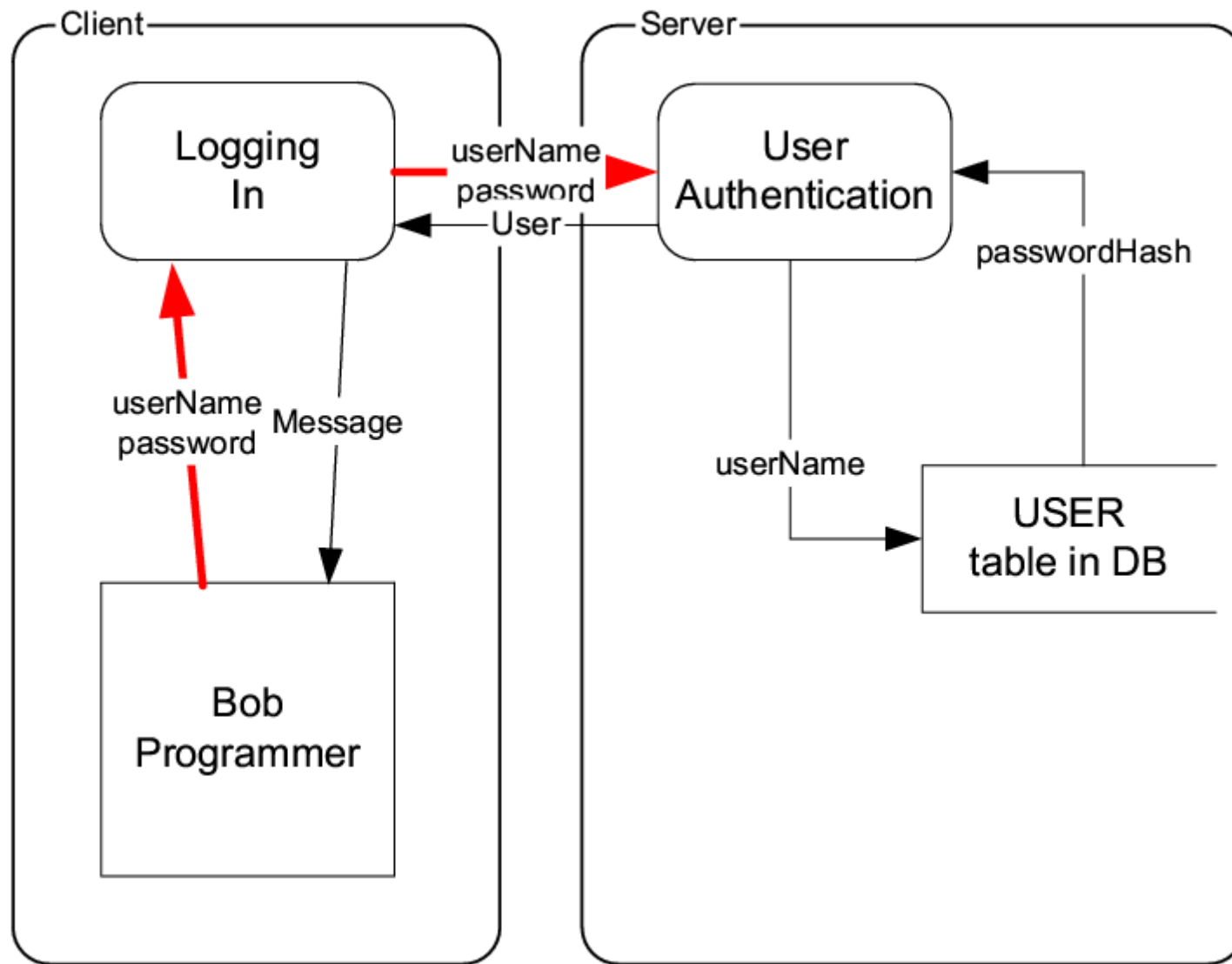
```
-defectId: int
-defectTypeNames: constant String Array
-defectTypeId: int
-phaseInjected: Phase
-phaseRemoved: Phase
-fixTime: int
-defectDescription: String
-defectStatus: bool

+getPhaseInjected(): Phase
+setPhaseInjected(phs:Phase): bool
+getPhaseRemoved(): Phase
+setPhaseRemoved(phs:Phase): bool
+getFixTime(): int
+setFixTime(time:int): bool
+getDefectTypeId(): int
+setDefectTypeId(time:int): int
+getDefectDescription(): String
+setDefectDescription(defdesc:String): bool
+getDefectStatus(): bool
+setDefectStatus(stat:bool): bool
```

# Sequence Diagram. "fillPIP"



# Data Flow Diagram. "UserAuth"





More Questions?