

# DD1346 - Javaprojekt - Projektplan

Carl Svensson, F-10

Carl Aronsson, F-10

January 31, 2013

## 1 Översikt

Chatklienten är strukturerad enligt MVC-mönstret. View-delen består av ett flertal klasser som representerar fönster och paneler. Controller-delen är delad i två klasser, en klass som sköter generella saker för hela programmet och en klass som har ett objekt för varje chatt. Model-delen har en klass för inställningar och tillstånd som gäller hela programmet och en klass som har ett objekt per chatt.

## 2 Klasser

### 2.1 View

#### **ProgramView**

Huvuprogrammet, innehåller en menyrad, en ruta med flikar innehållandes instanser av ChatView samt en MessageComposerView.

#### **ChatView**

Representerar en chat. Visar en lista med chattens deltagare och alla meddelanden som skrivits.

#### **MessageComposerView**

Textredigeraren, innehåller en textruta samt alternativ för att formatera texten och välja kryptering.

#### **AboutBoxView**

Dialogruta som öppnas när man trycker på Help → About. Visar vilka som tillverkat programmet samt versionsinformation.

#### **SettingsView**

Ger användaren möjlighet att visa och redigera inställningar så som lyssningsport och namn.

**FileTransferStartView**

Dialog för att påbörja en filöverföring. Låter användaren välja en fil och skriva ett meddelande.

**FileTransferRequestView**

Dialogruta för inkommande filöverföringsförfrågan. Visar filnamnet och meddelandet samt låter användaren välja om den ska ta emot den.

**FileTransferView**

Dialogruta som visar status på en pågående filöverföring.

**NewChatView**

Dialogruta för att skapa en ny chatt. Låter användaren bestämma om den ska ansluta till en annan chat eller skapa en gruppchatt.

## 2.2 Controller

**ProgramCtrl**

Programkontrollen, sköter globala inställningar, öppnande och stängande av dialogrutor och skapande av nya chattar.

**ProgramCtrl**

Chattkontrollen, sköter en chattsession. Tar emot och skickar meddelanden till en eller flera klienter.

## 2.3 Model

**ProgramSettingsMdl**

Lagrar inställningar för programmet så som port och namn. Låter även programmet spara dessa och läsa in dem från en fil.

**ChatMdl**

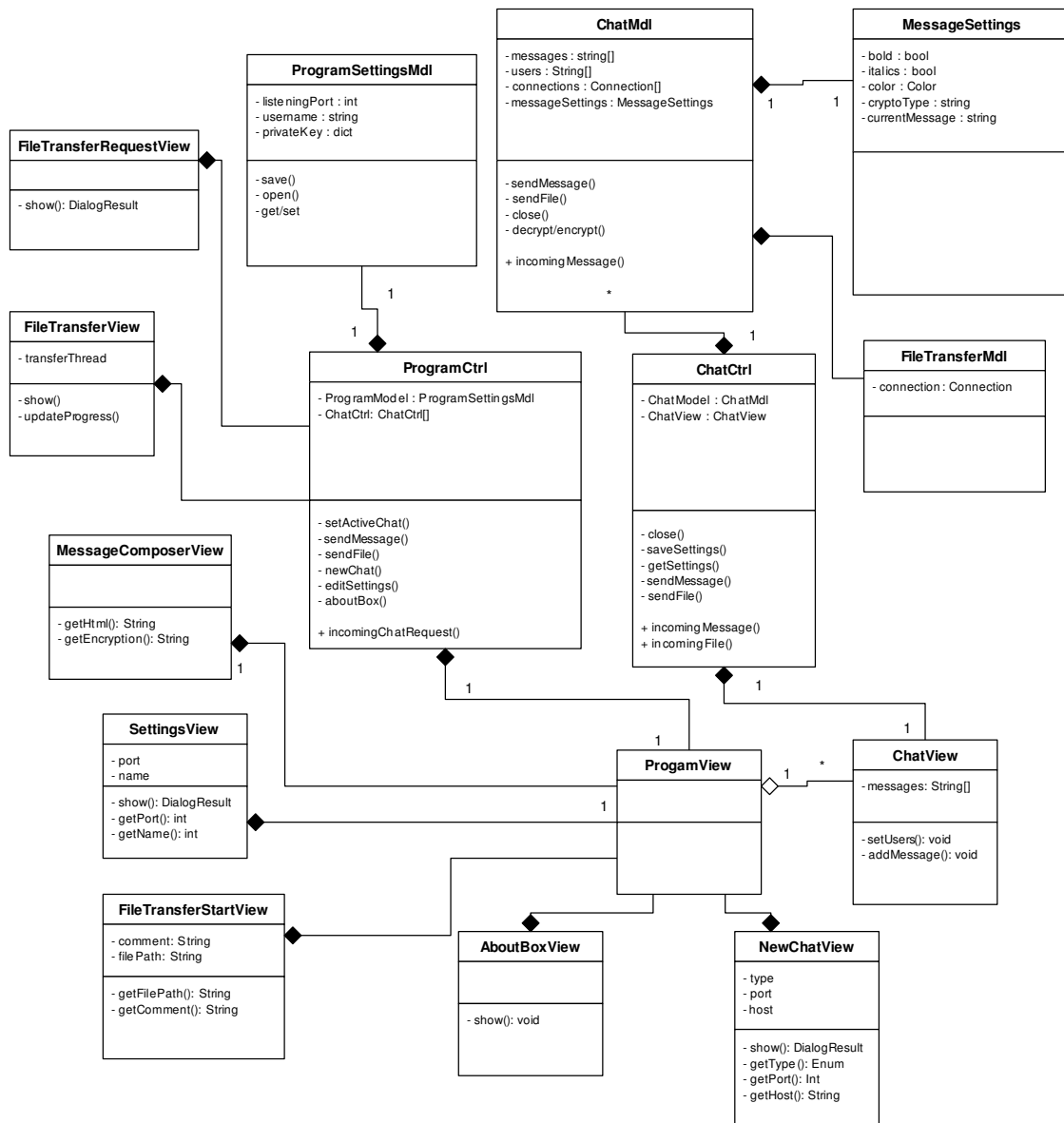
Chatmodell, innehåller anslutningar och användare i en chatt. Lagrar även meddelandeinställningar i ett MessageSetting-objekt. Ansvarar även för initiering av filöverföring samt kryptering.

**MessageSettings**

Meddelandeinställningar, lagrar inställningar som skickar till MessageComposerView när chatten blir den aktiva. Detta eftersom det bara finns ett sådant objekt men flera chattar.

### 3 UML

Nedan visas programmetts UML-klassdiagram med klassernas viktigaste funktioner samt hur kompositionen ser ut.



## 4 Kontrollflöden

Nedan beskrivs hur kontrollen flödar mellan klasserna under de olika användningsfallen.

### 4.1 Ändra inställningar

Användaren trycker på Edit → Settings. *ProgramView* skapar ett *SettingsView*-objekt och visar det. När användaren stänger dialogrutan kontrolleras om denne sparade och skickar i så fall resultatet till *ProgramCtrl* som uppdaterar *ProgramSettingsMdl*.

### 4.2 Ansluta till chat

Användaren trycker på File → New chat. *ProgramView* skapar ett *NewChatView*-objekt och tittar sedan på resultaten. Detta skickas till *ProgramCtrl* som skapar ett *ChatView*-objekt som i sin tur skapar ett *ChatCtrl*-objekt som i sin tur skapar ett *ChatMdl*-objekt. Detta ansluter till värden och börjar kommunicera. *ChatView*-objektet läggs till *ProgramView*-objektet.

### 4.3 Ta emot inkommande anslutning

Programmet får en ny anslutning i *ProgramCtrl*. Den frågar användaren om den vill acceptera anslutningen och skapar i så fall en ny chatt på samma sätt som i förgående avsnitt.

### 4.4 Skapa gruppchat

Användaren trycker på File → New chat. Precis som tidigare så skapar *ProgramView* en dialogruta vars resultat skickas till *ProgramCtrl* som skapar en ny chat som tidigare men som lyssnar på en ny port för anslutningar som sedan kan läggas till i *ChatMdl*-objektet.

### 4.5 Skicka fil

Användaren trycker på Send File. *ProgramView* skapar ett *FileTransferStartView*-objekt vars resultat skickas via *ChatView* och *ChatCtrl* till *ChatMdl* som skickar ett meddelande till motparten om filöverföring.

## 4.6 Ta emot fil

*ChatMdl* tar emot ett meddelande om filöverföring vilket skickas till *ProgramCtrl* som, genom ett *FileTransferRequestView*-objekt frågar användaren om den vill ta emot filen. Om ja skickas ett meddelande till avsändaren som öppnar en ny anslutning för filen. Detta sparas i ett *FileTransferMdl*. Både avsändare och mottagare skapar ett *FileTransferView*-objekt som övervakar filöverföringens status.