Requirement Report

Chinese Chess
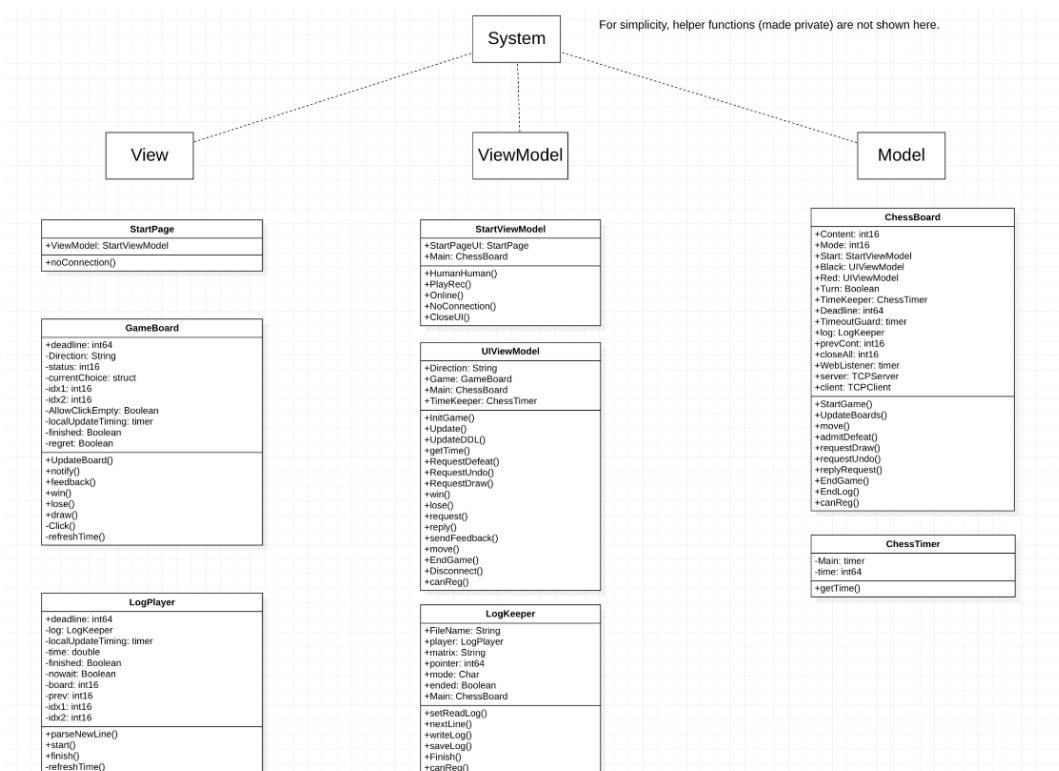
Team3

Author(R): XXXX

Content

Introductions & Rules

This is an app which user can play Chinese chess locally or online on.
Basic knowledge of Chinese chess:
1. Pieces: There are 7 types: King, Advisor, Elephant, Rook, Cannon, Horse, and Pawn.
2. Rules: Pieces can move and capture according to their individual ability and restriction along the lines including the "Riverbanks". It is noteworthy that it has some limitations such as King, Advisors must move inside their own Palaces, Horses and Elephants can be blocked (from moving), two King cannot "face" each other (stands in the same column without any piece between). Moreover, 3-FOLK rules need to be followed: when a position repeated 3 time (3 folks) the game will be terminated.

System Structure

For simplicity, helper functions (made private) are not shown here.

System

View — **ViewModel** — **Model**

StartPage
+ViewModel: StartViewModel
+noConnection()

GameBoard
+deadline: int64
-Direction: String
-status: int16
-currentChoice: struct
-idx1: int16
-idx2: int16
-AllowClickEmpty: Boolean
-localUpdateTiming: timer
-finished: Boolean
-regret: Boolean
+UpdateBoard()
+notify()
+feedback()
+win()
+lose()
+draw()
-Click()
-refreshTime()

LogPlayer
+deadline: int64
-log: LogKeeper
-localUpdateTiming: timer
-time: double
-finished: Boolean
-nowait: Boolean
-board: int16
-prev: int16
-idx1: int16
-idx2: int16
+parseNewLine()
+start()
+finish()
-refreshTime()

StartViewModel
+StartPageUI: StartPage
+Main: ChessBoard
+HumanHuman()
+PlayRec()
+Online()
+NoConnection()
+CloseUI()

UIViewModel
+Direction: String
+Game: GameBoard
+Main: ChessBoard
+TimeKeeper: ChessTimer
+InitGame()
+Update()
+UpdateDDL()
+getTime()
+RequestDefeat()
+RequestUndo()
+RequestDraw()
+win()
+lose()
+request()
+reply()
+sendFeedback()
+move()
+EndGame()
+Disconnect()
+canReg()

LogKeeper
+FileName: String
+player: LogPlayer
+matrix: String
+pointer: int64
+mode: Char
+ended: Boolean
+Main: ChessBoard
+setReadLog()
+nextLine()
+writeLog()
+saveLog()
+Finish()
+canReg()

ChessBoard
+Content: int16
+Mode: int16
+Start: StartViewModel
+Black: UIViewModel
+Red: UIViewModel
+Turn: Boolean
+TimeKeeper: ChessTimer
+Deadline: int64
+TimeoutGuard: timer
+log: LogKeeper
+prevCont: int16
+closeAll: int16
+WebListener: timer
+server: TCPServer
+client: TCPClient
+StartGame()
+UpdateBoards()
+move()
+admitDefeat()
+requestDraw()
+requestUndo()
+replyRequest()
+EndGame()
+EndLog()
+canReg()

ChessTimer
-Main: timer
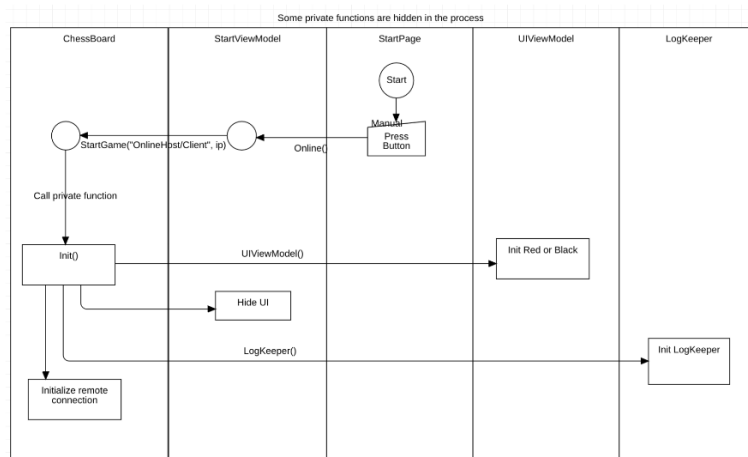-time: int64
+getTime()

Software Requirement

R1 Main menu(start game)

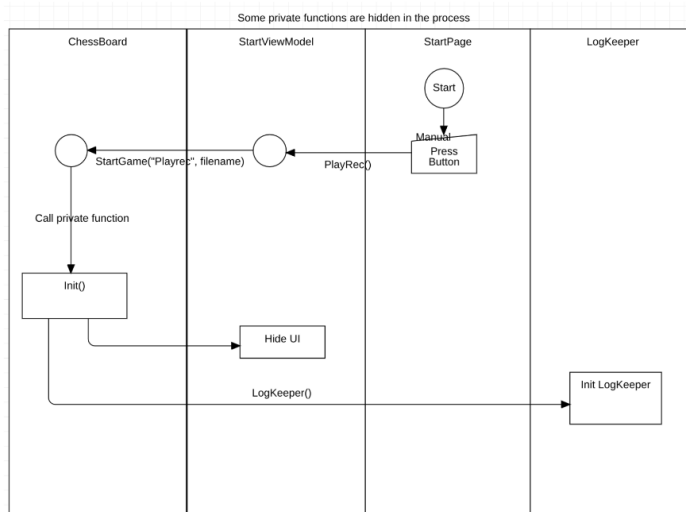R1.1: There is a main menu GUI after entering the game.

R1.2: User can select the local game on the main menu.
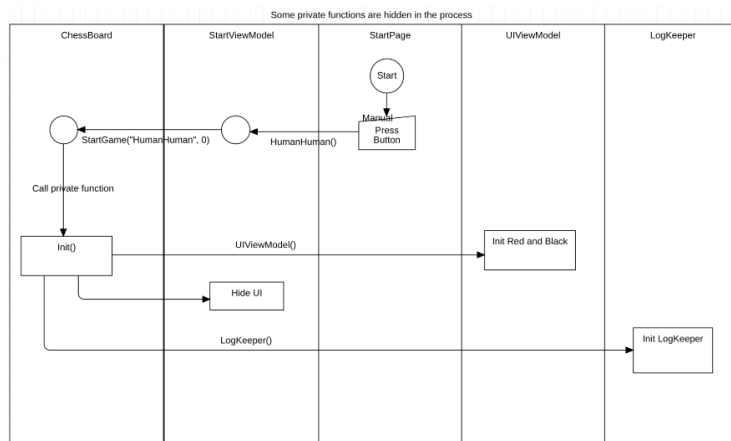
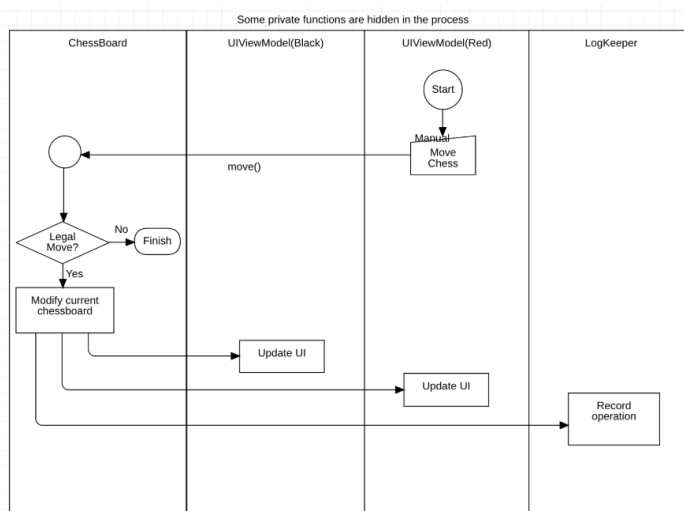R1.3: User can select the online game on the main menu.

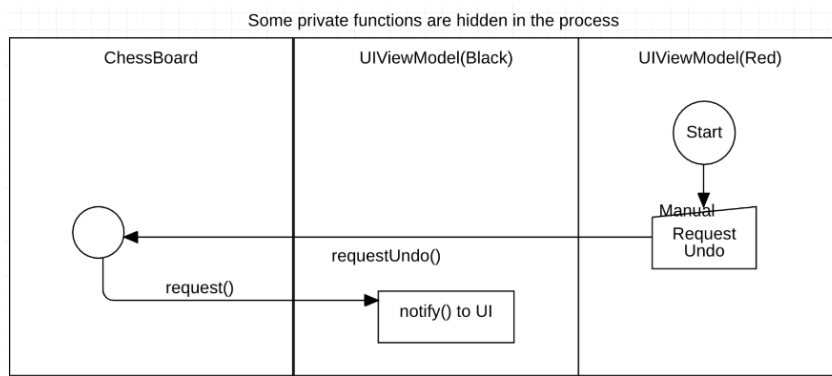R1.4: User can select to watch recordings on the main menu.

Some private functions are hidden in the process

R2 Local game

R2.1: There are 2 gaming GUIs which contain chess board, chess pieces, some functional buttons and left time. The components on the GUIs should be displayed properly.
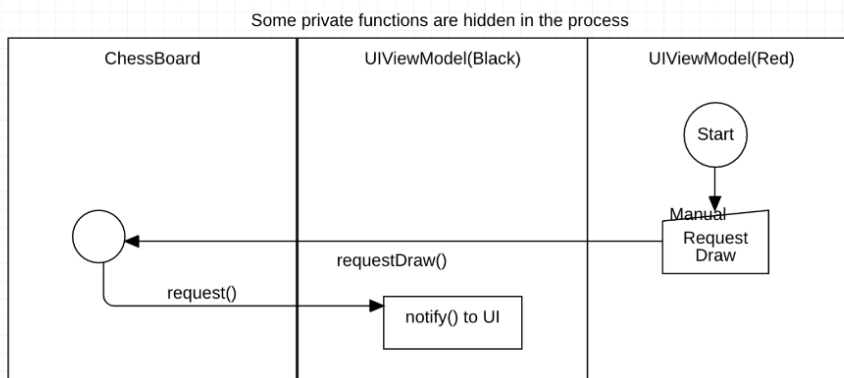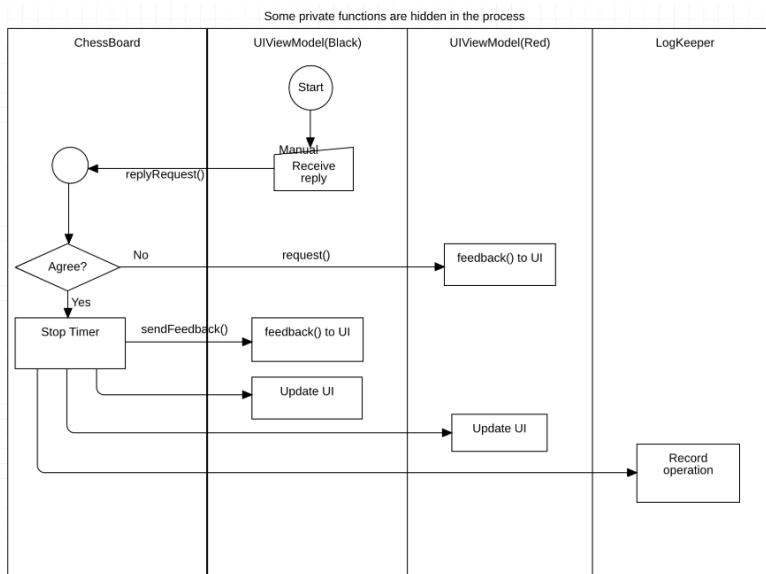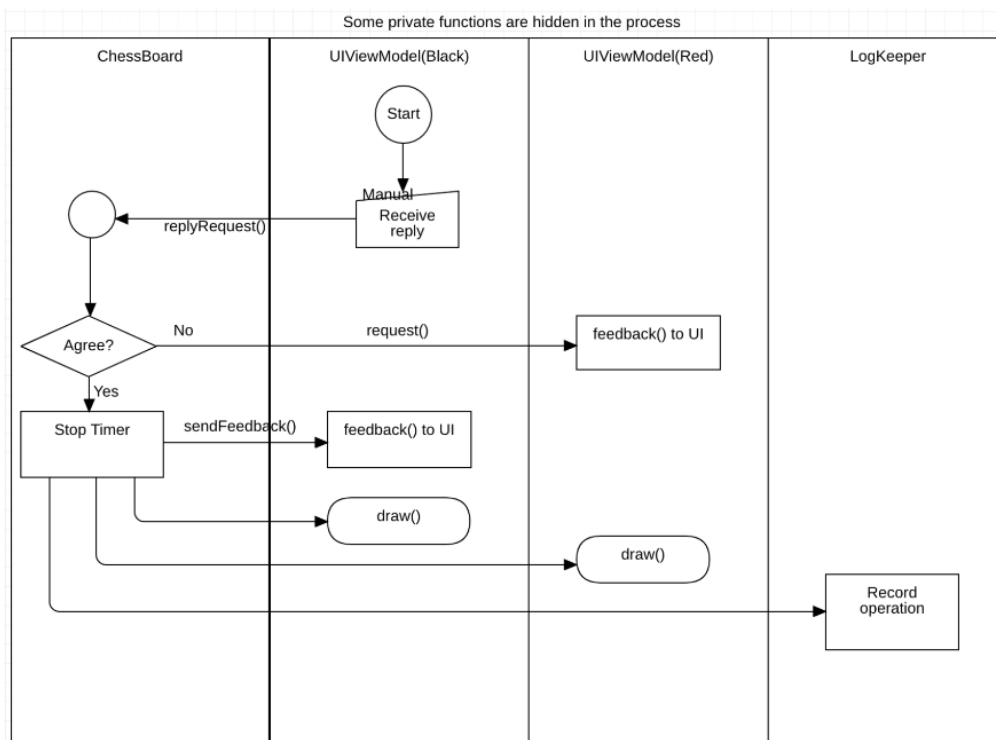
R2.2: User can move chess.

Some private functions are hidden in the process

R2.3: User can request for undo.

Some private functions are hidden in the process

ChessBoard	UIViewModel(Black)	UIViewModel(Red)

Start

Manual
Request
Undo

requestUndo()

request()

notify() to UI

R2.4: User can request for draw.

Some private functions are hidden in the process

ChessBoard	UIViewModel(Black)	UIViewModel(Red)

Start

Manual
Request
Draw

requestDraw()

request()

notify() to UI

R2.5: User can agree/deny undo.

ChessBoard	UIViewModel(Black)	UIViewModel(Red)	LogKeeper

Start

Manual
Receive reply

replyRequest()

Agree?

No

request()

feedback() to UI

Yes

Stop Timer

sendFeedback()

feedback() to UI

Update UI

Update UI

Record operation

R2.6: User can agree/deny draw.

ChessBoard	UIViewModel(Black)	UIViewModel(Red)	LogKeeper

Start

Manual
Receive reply

replyRequest()

Agree?

No

request()

feedback() to UI

Yes

Stop Timer

sendFeedback()

feedback() to UI

draw()

draw()

Record operation

R2.7: User can surrender.

Some private functions are hidden in the process
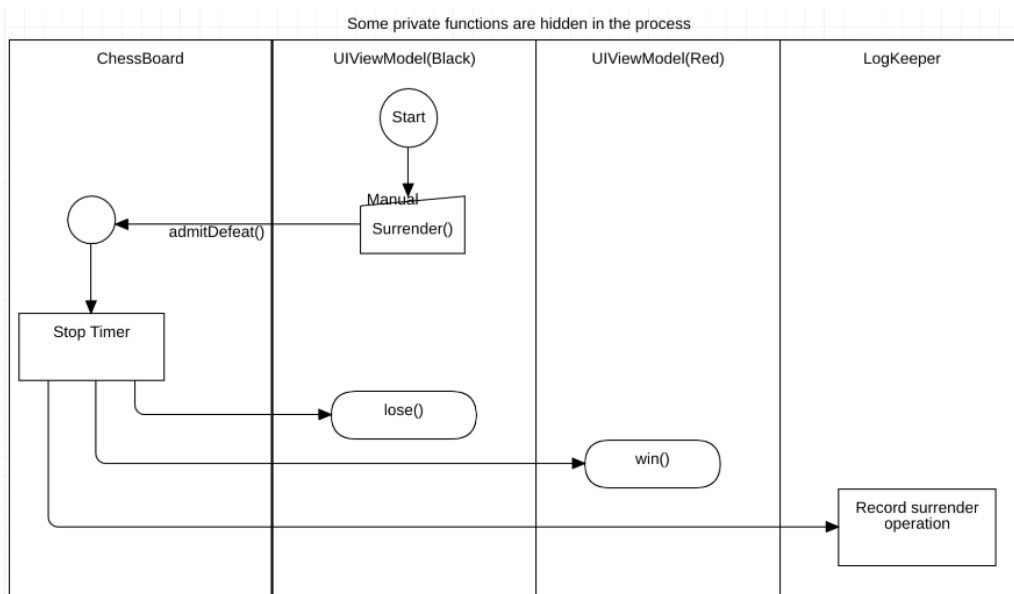
R2.8: In each turn, time is limited.

R2.9: The system should be able to determine whether the player wins or whether the game is a draw.

R2.10: The two players can't move at the same time.

R2.11: The 2 GUIs for two players should be updated in each turn.

R2.12: The whole game should obey the basic rules of Chinese chess.

R3 Online game

**Note: R3 should satisfy all the requirements in R2.**

R3.1: There are host and remote client. There are 2 GUIs for them, one is local and another is remote.

R3.2: The host and remote client can play the game together under the LAN.

R4 Records

R4.1: User can watch each game's recording.

R4.2: The game's recording can be displayed automatically.

Some private functions are hidden in the process
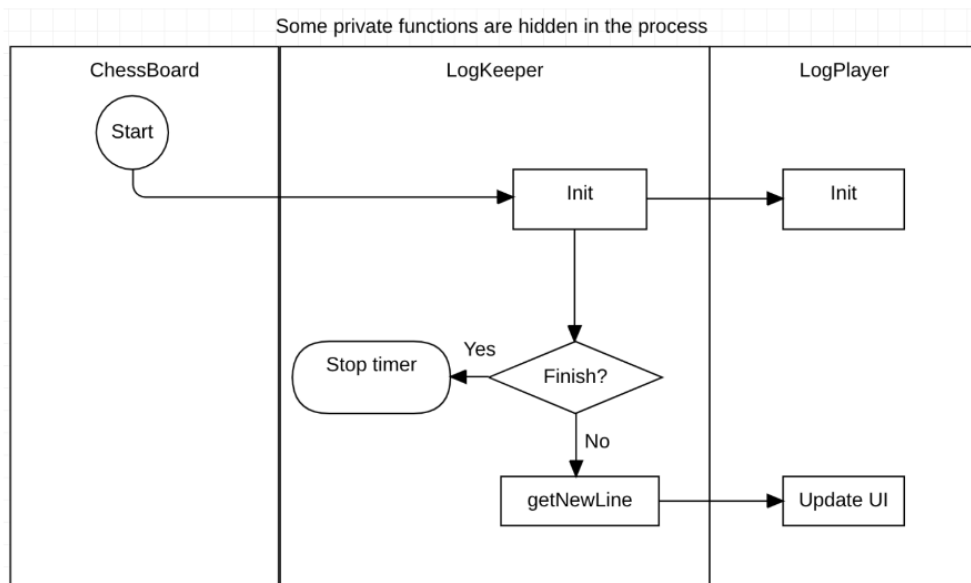
R4.3: The user can set the speed of recordings.