Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24.03.2014 | 1.0 | Add GUI components’ design | Erdi Gültekin |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

1 Introduction 3

1.1 References 3

1.1.1 Project References 3

2 Software Architecture overview 3

3 Software design description 3

3.1 Component 1 3

3.1.1 Component interfaces 3

3.1.2 Component design description 3

3.1.3 Workflows and algorithms 3

3.1.4 Software requirements mapping 3

3.2 Component 2 3

3.2.1 Component interfaces 3

3.2.2 Component design description 4

3.2.3 Workflows and algorithms 4

3.2.4 Software requirements mapping 4

3.3 Component 3 4

3.3.1 Component interfaces 4

3.3.2 Component design description 4

3.3.3 Workflows and algorithms 4

3.3.4 Software requirements mapping 4

4 COTS Identification 4

# Introduction [EREN]

Brief description of the software system and the purpose of the document.

This document describes the design of the XXX software system.

## References

### Project References

| # | Document Identifier | Document Title |
| --- | --- | --- |
| [R1] | ID | Add your documents references.  One line per document |

# Software Architecture overview [EREN]

Describe here the top level software components and their interactions/relationships.

Use UML diagrams.

# Software design description

Describe each top level package/component of your software and if necessary sub-components/sub packages.

Use Class diagrams, sequence diagrams and deployment diagrams to illustrate your description.

## Graphical User Interface [ERDI]

### Component interfaces

The user interface of TicTacToe game consists of 4 different components which are MainMenuPanel, NetworkMenuPanel, GameBoard and BoardButton.

### Component design description

### 

MainMenuPanel: This component creates the main menu of the game. It has two buttons to redirect user either to a local game or to a network game.  
  
NetworkMenuPanel: This component can be triggered by network game button on the MainMenuPanel. It presents host game and join game buttons to user in order to set up a network game. Join game option uses pop-up box to get IP address information. Host game option uses pop-ups to inform the user about status of remote play (e.g. player is connecting).

GameBoard: This component can be triggered by local game, join game or host game buttons. It creates the game board which has 9 “BoardButton”s and a player turn indicator. It is also creates pop-up boxes to inform the user about game results and the status of remote player (e.g. disconnection).

BoardButton: This component is used by GameBoard. It includes fields that are required for game such as its status and its sign.

### Workflows and algorithms

The algorithms needed for TicTacToe are handled by Logic component. Please see Logic section.

### Software requirements mapping

SRS-REQ-101: This requirement is handled by MainMenuPanel, GameBoard, BoardButton and relevant other Controller & Logic units.

SRS-REQ-102: This requirement is handled by MainMenuPanel, NetworkMenuPanel GameBoard, BoardButton and relevant other Controller & Logic units.

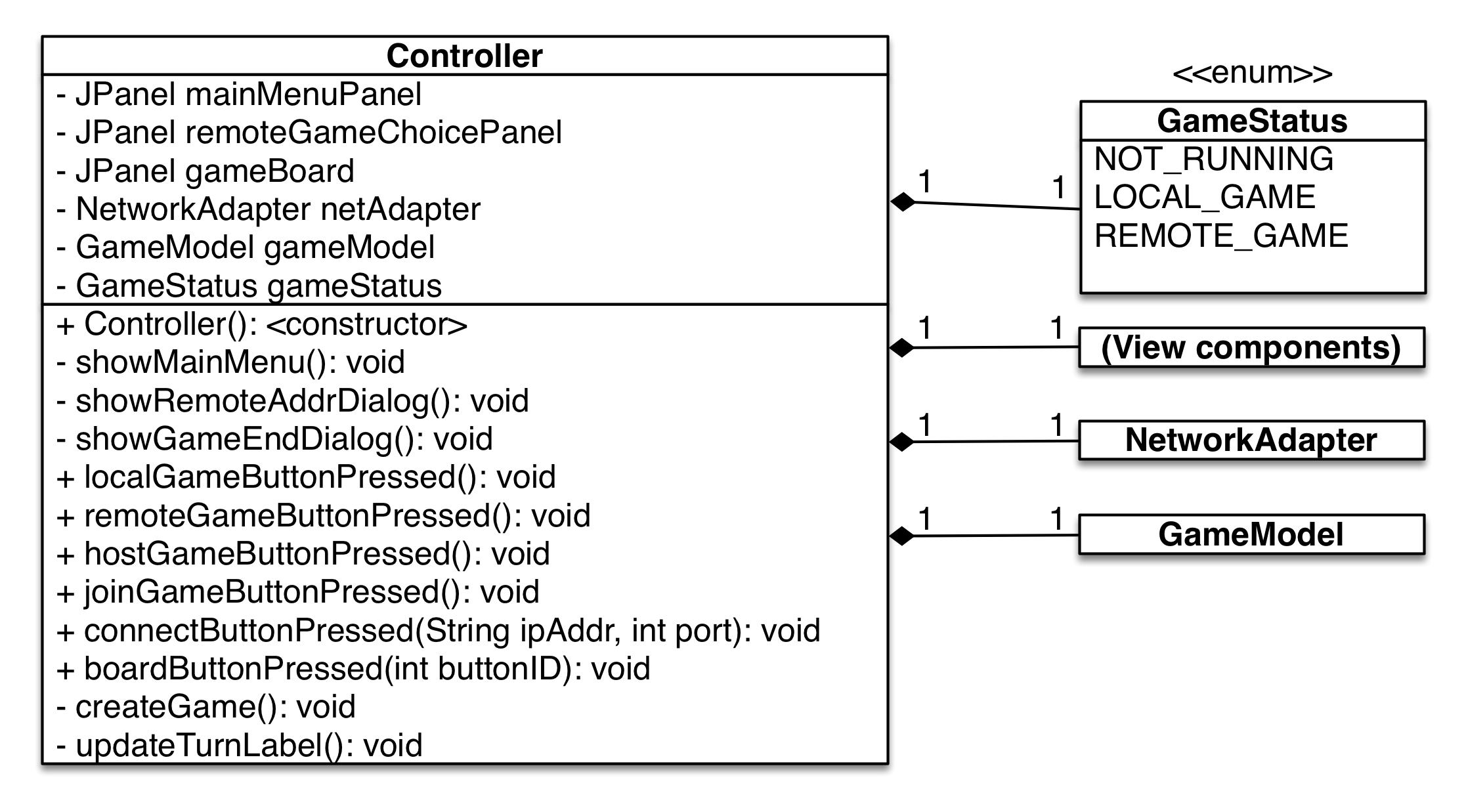
SRS-REQ-103: This requirement is handled by MainMenuPanel, NetworkMenuPanel GameBoard, BoardButton and relevant other Controller & Logic units.

## Controller

### Component interfaces

Describe the interfaces of the component and input output data

### Component design description



### Macintosh HD:Users:aemreunal:Documents:GitHub:CS320-Project1-TicTacToe:Class UML diagrams:Controller workflow.png Workflows and algorithms

### Software requirements mapping

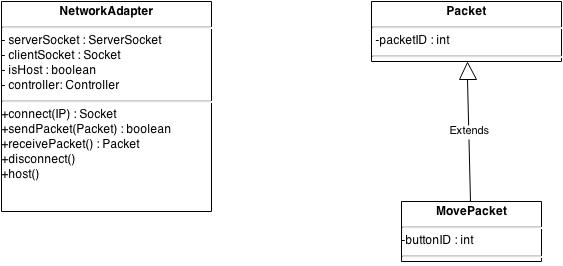
None.

## Network [DENIZ]

### Component interfaces

The Network component basically consists of two parts: Network Adapter and the Network Packets.

### Component design description



NetworkAdapter: Controls all of the network flow and provides all network-related actions in the game. It can host or connect a game, send packets through the network to the peer or listen to the peer to receive a packet.

Packet: Serializable network packets which are serialized and sent through the network and deserialized on receive. Carries information about the action.

### Workflows and algorithms

Use sequence diagrams to show the workflows of components/packages/classes inside the component.

Describe algorithms, if possible. An algorithm may be described outside this document, in this case, add the reference to that document.

### Software requirements mapping

List the SRS requirements handled by this component

## Logic [Eren]

Repeat the pattern for each component.

### Component interfaces

Describe the interfaces of the component and input output data

### Component design description

Describe the design of the component, Use class diagrams to show the links between sub-components/sub-packages and or classes inside the component.

### Workflows and algorithms

Use sequence diagrams to show the workflows of components/packages/classes inside the component.

Describe algorithms, if possible. An algorithm may be described outside this document, in this case, add the reference to that document.

### Software requirements mapping

List the SRS requirements handled by this component

# COTS Identification [ALL]

List external software components/libraries that your system rely on, if there are any.

Example:

COTS (commercial of the shelf) libraries used in XXX are the following:

* foo.jar, version id, download URL, License type,
* bar.jar, version id, download URL, License type,