<u>Table of Contents:</u>
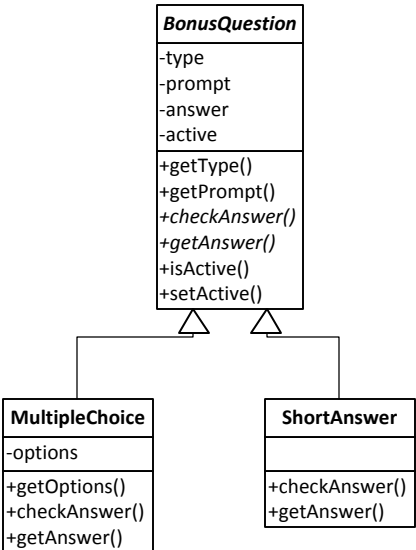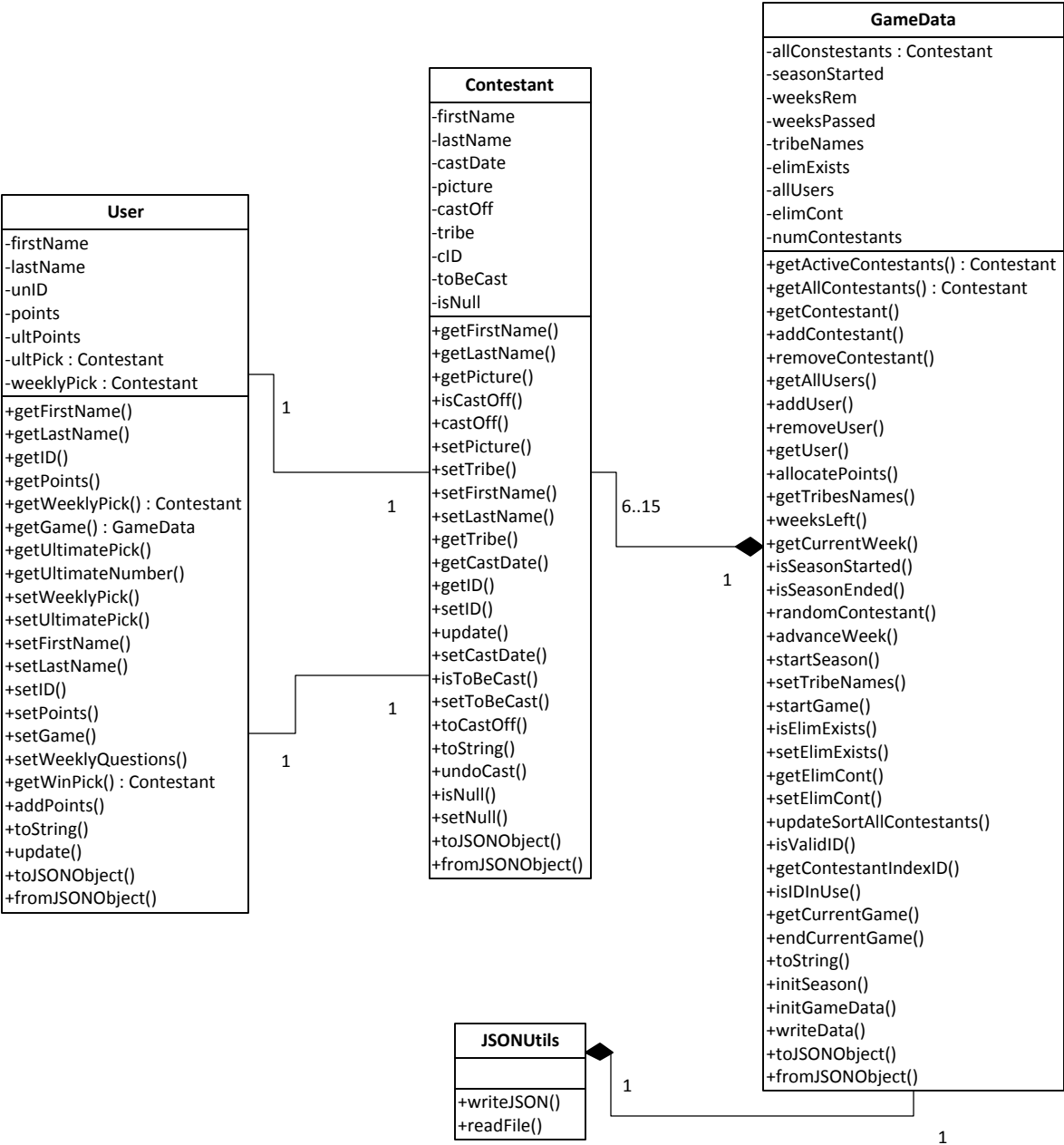
<u>Section 1:</u>

- UML Class Diagram

<u>Section 2:</u>

- Testing
  - Test case listing
  - Driver and stub examples

<u>Section 3:</u>

- Project Plan (Gantt Chart)

User

-firstName
-lastName
-unID
-points
-ultPoints
-ultPick : Contestant
-weeklyPick : Contestant

+getFirstName()
+getLastName()
+getID()
+getPoints()
+getWeeklyPick() : Contestant
+getGame() : GameData
+getUltimatePick()
+getUltimateNumber()
+setWeeklyPick()
+setUltimatePick()
+setFirstName()
+setLastName()
+setID()
+setPoints()
+setGame()
+setWeeklyQuestions()
+getWinPick() : Contestant
+addPoints()
+toString()
+update()
+toJSONObject()
+fromJSONObject()

Contestant

-firstName
-lastName
-castDate
-picture
-castOff
-tribe
-cID
-toBeCast
-isNull

+getFirstName()
+getLastName()
+getPicture()
+isCastOff()
+castOff()
+setPicture()
+setFirstName()
+setLastName()
+getTribe()
+getCastDate()
+getID()
+setID()
+update()
+setCastDate()
+isToBeCast()
+setToBeCast()
+toCastOff()
+toString()
+undoCast()
+isNull()
+setNull()
+toJSONObject()
+fromJSONObject()

GameData

-allContestants : Contestant
-seasonStarted
-weeksRem
-weeksPassed
-tribeNames
-elimExists
-allUsers
-elimCont
-numContestants

+getActiveContestants() : Contestant
+getAllContestants() : Contestant
+getContestant()
+addContestant()
+removeContestant()
+getAllUsers()
+addUser()
+removeUser()
+getUser()
+allocatePoints()
+getTribesNames()
+weeksLeft()
+getCurrentWeek()
+isSeasonStarted()
+isSeasonEnded()
+randomContestant()
+advanceWeek()
+startSeason()
+setTribeNames()
+startGame()
+isElimExists()
+setElimExists()
+getElimCont()
+setElimCont()
+updateSortAllContestants()
+isValidID()
+getContestantIndexID()
+isIDInUse()
+getCurrentGame()
+endCurrentGame()
+toString()
+initSeason()
+initGameData()
+writeData()
+toJSONObject()
+fromJSONObject()

JSONUtils

+writeJSON()
+readFile()

1   1

6..15

1

1

1

1

BonusQuestion

-type
-prompt
-answer
-active

+getType()
+getPrompt()
+*checkAnswer()*
+*getAnswer()*
+isActive()
+setActive()

MultipleChoice

-options

+getOptions()
+checkAnswer()
+getAnswer()

ShortAnswer

+checkAnswer()
+getAnswer()

ID: 1

TESTING: **Admin User creating a season with no tribe names**

REFERENCES: Requirement 4.c.1.2

TEST CASE:

1. At unix command prompt, run admin.MainFrame

2. At opening screen, do not enter tribe names

3. Click "create season" at the bottom of the panel

EXPECTED OUTPUT:

Error --> Message: *Invalid tribe names*

ID: 2

TESTING: **Admin User creating a season with invalid (too long) tribe names**

REFERENCES: Requirement 4.c.1.2

TEST CASE:

1. While on the Create Season panel

2. Enter any string of characters with a length > 30 in one tribe name

3. Enter "abcde" in the second tribe panel

EXPECTED OUTPUT:

Error --> Message: *Invalid tribe name*

ID: 3

TESTING: **Admin User creating a season with special characters in tribe names**

REFERENCES: Requirement 4.c.1.2

TEST CASE:

1. While on the create season panel

2. Enter "wxyz123" as one of the tribe names

3. Enter "abcde" as the second tribe name

EXPECTED OUTPUT:

Error --> Message: *Invalid tribe name*

ID: 4

TESTING: **Admin User creating new season**

REFERENCES: Requirement 4.c.1.1

TEST CASE:

1. On the Create Season panel

2. Enter "wxyz" as the first tribe name

3. Enter "abcde"as the second tribe name

EXPECTED OUTPUT:

Successful season create -----> Panel should switch to general panel

ID: 5

TESTING: **Admin User changing themes of the GUI**

REFERENCES: Requirement 4.c.1.1

TEST CASE:

1. On any panel, select "Theme" from the taskbar

2. Select any of the themes (default should be "Snow")

EXPECTED OUTPUT:

Background/buttons should successfully change to chosen theme

ID: 6

TESTING: **Admin User saving a new contestant with invalid id**

REFERENCES: Requirement 4.c.1.2

TEST CASE:

1. Enter the contestant panel

2. Enter "wxyz" into the first name field, and "abcde" in the last name field

3. Enter "abcde" for user id and press "save"

EXPECTED OUTPUT:

Error --> Message: *Invalid user id*

ID: 7

TESTING: **Admin User saving a new contestant with invalid name**

REFERENCES: Requirement 4.c.1.2

TEST CASE:

1. While on the contestant panel

2. Enter "wzyx123" for the first name, and "abcde" as the last name

3. Enter *"wa"* as the Contestant ID, and press "Save"

EXPECTED OUTPUT:

Error --> Message: *Invalid First Name, (must be alphabetic, 1-20 characters*

ID: 8

TESTING: **Admin User saving a new contestant**

REFERENCES: Requirement 4.c.1.2

TEST CASE:

1. While on the contestant panel

2. Enter "wxyz" as the first name, and "abcde" as the last name

3. Enter "wa" for the Contestant ID, and press "Save"

EXPECTED OUTPUT:

Contestant should be successfully added to the panel/game.

ID: 9

TESTING: **Admin User editing existing contestant**

REFERENCES: Requirement 4.c.1.5

TEST CASE:

1. With newly created contestant selected in the table

2. Switch the first name with the last name, and change the id to "aw"

3. Press "Save"

EXPECTED OUTPUT:

Contestant should update appropriately in table

ID: 9

TESTING: **Admin User starting season with wrong number of contestants**

REFERENCES: No specific requirement

TEST CASE:

1. After adding 0 – (weeks-2) contestants, go to the General panel

- Note – to add a new contestant, you must press "Add New" at the bottom, otherwise you will continuously edit an existing contestant.

2. Press "Start season"

EXPECTED OUTPUT:

Error --> Message: *You must enter x contestants to start.*

ID: 10

TESTING: **Admin User starting season with correct number of contestants**

REFERENCES: Requirement 4.c.6

TEST CASE:

1. After adding weeks-3 contestants, go to the General panel

2. Press "Start Season"

EXPECTED OUTPUT:

Input Dialog should come up, asking for weekly bet amount

ID: 11

TESTING: **Admin User entering incorrect bet amount**

REFERENCES: Requirement 4.c.6

TEST CASE:

1. At the weekly dollar amount prompt (after starting season)

2. Enter anything other than integers

EXPECTED OUTPUT:

Error --> Message: *Invalid amount entered.*

ID: 12

TESTING: **Admin User casting off after season started**

REFERENCES: Requirement 4.c.8

TEST CASE:

1. After season has started, go to Contestant Panel

2. Select a user from the table

3. Press "Cast Off"

EXPECTED OUTPUT:

Table updates appropriately – label changes to note cast off

ID: 13

TESTING: **Admin User undoing that cast**

REFERENCES: Requirement 4.c.10

TEST CASE:

1. While still on Contestant Panel

2. Select another user, and then back to the user that was previously cast off

3. Click "Undo cast off"

EXPECTED OUTPUT:

Tables reset to appropriate values prior to initial cast off

ID: 14

TESTING: **Admin User casting off more than one contestant per week**

REFERENCES: No specific requirement

TEST CASE:

1. On the contestant panel

2. Select a user and press "Cast Off"

3. Select a second user and press "Cast Off"

EXPECTED OUTPUT:

Error --> Message: *Explaining how only one user can be cast off per week.*

ID: 15

TESTING: **Admin User undoing a cast from a previous week**

REFERENCES: Requirement 4.c.10

TEST CASE:

1. After selecting a contestant to be cast off, go to the General Panel

2. Press "Advance week", and return to the contestant panel

3. Select the contestant that was cast off, and press "Undo Cast"

EXPECTED OUTPUT:

All changes made to the table from "Cast Off" should be undone

ID: 16

TESTING: **Admin User attempting to advance week without casting**

REFERENCES: No specific requirement

TEST CASE:

1. Without selecting a Contestant to be cast off, go to the General Panel

2. Press "Advance Week"

EXPECTED OUTPUT:

Error --> Message: *Contestant must be selected to be cast off.*

SECTION: ADMIN USER ADDING BONUS QUESTIONS

ID: 17

TESTING: **Admin User adding a multiple choice question correctly**

REFERENCES: Requirement 4.c.9

TEST CASE:

1. Go to the Bonus Panel

2. Select Multiple Choice question, and press "Add New Question"

3. Enter the question and 3-4 possible answers

4. Indicate which answer is correct, and press "Save Question"

EXPECTED OUTPUT:

Question added properly → Message: Question saved!

ID: 18

TESTING: **Admin User adding multiple choice question, no answer indicated**

REFERENCES: Requirement 4.c.9

TEST CASE:

1. Go to the Bonus Panel

2. Select multiple choice question, and press "Add New Question"

3. Enter question and 3-4 answers, but **do not** indicate which is the answer

4. Press "Save Question"

EXPECTED OUTPUT:

Error → JMessageDialog: *"You must indicate which is the proper answer."*

ID: 19

TESTING: **Admin User adding multiple choice question, no question**

REFERENCES: Requirement 4.c.9

TEST CASE:

1. Go to the Bonus Panel

2. Select multiple choice question, and press "Add New Question"

3. Enter no question and some answers (or no answers)

4. Press "Save Question"

EXPECTED OUTPUT:

Error → JMessageDialog: *"No question to be saved."*

ID: 20

TESTING: **Admin User adding short answer question correctly**

REFERENCES: Requirement 4.c.9

TEST CASE:

1. Go to the Bonus Panel

2. Select short answer question, and press "Add New Question"

3. Enter a question in the question field, and an answer in the answer field

4. Press "Save Question"

EXPECTED OUTPUT:

Question added properly → Message: Question Saved!

ID: 21

TESTING: **Admin User adding short answer question, no answer**

REFERENCES: Requirement 4.c.9

TEST CASE:

1. Go to the Bonus Panel

2. Select short answer question, and press "Add New Question"

3. Enter a question in the question field, and leave the answer field blank

4. Press "Save Question"

EXPECTED OUTPUT:

Error → JMessageDialog: *"There must be an answer to every question."*

ID: 22

TESTING: **Admin User adding short answer question, no question**

REFERENCES: Requirement 4.c.9

TEST CASE:

1. Go to the Bonus Panel

2. Select short answer question, and press "Add New Question"

3. Leave the question field blank, and type an answer in the answer field

4. Press "Save Question"

EXPECTED OUTPUT:

Error → JMessageDialog: *"You must enter a question to be asked!"*

SECTION: CLIENT SELECTING WINNER OF COMPETITION

ID: 23

TESTING: **Client User selecting which contestant will win the competition**

REFERENCES: Requirement 4.d.1.3

TEST CASE:

1. Log in to the BB application

2. Select Contestant from the "Ultimate Pick" drop down box

3. Press "Save Selection"

EXPECTED OUTPUT:

Selection successfully saved → Message: Selection saved!

ID: 24

TESTING: **Client User updating which contestant will win the competition**

REFERENCES: Requirement 4.d.1.3

TEST CASE:

1. Log in to the BB application

2. Select Contestant from the "Ultimate Pick" drop down box

3. Press "Save Selection"

EXPECTED OUTPUT:

Selection successfully saved → Message: Selection saved!

ID: 25

TESTING: **Client User selecting the null selection for which contestant will win**

REFERENCES: Requirement 4.d.1.3

TEST CASE:

1. Log in to the BB application

2. Select "[??] Null" from the "Ultimate Pick" drop down box

3. Press "Save Selection"

EXPECTED OUTPUT:

JConfirmationDialog: "Saving will assign you a random contestant. Proceed?"

Section 2:

Driver and Stub Example:

```java
package test;

import java.util.ArrayList;

import data.Contestant;
import data.GameData;
import data.InvalidFieldException;


/*
 * Test Case: Admin casting off a contestant after season started
 */
public class Testing{

    //Constant values for testing purposes
    private static final Integer NUM_TEST_CONTESTANTS = 6;
    private static final Integer TEST_BET_AMOUNT = 5;
    private static final String TEST_TRIBE_ONE = "Alive";
    private static final String TEST_TRIBE_TWO = "Dead";


    //DRIVER TESTS TO SEE IF CAST OFF IS WORKING PROPERLY
    public static void driver() throws InvalidFieldException{

        //Instantiate a new GameData object with 6 test contestants.
        GameData g = new GameData(NUM_TEST_CONTESTANTS);

        //Instantiate valid tribe names for the game.
        g.setTribeNames(TEST_TRIBE_ONE, TEST_TRIBE_TWO);

        //Strings to be used to compose test Contestants, so that a season may
be started.
        String[] testID = {"MJ", "BP", "JS", "AJ", "AE", "JB"};
        String[] testFirstName = {"Michael", "Brad", "Jessica", "Angelina",
"Albert", "Al"};
        String[] testLastName = {"Jackson", "Pitt", "Simpson", "Jolie",
"Einstein", "Capone"};
        String[] testTribe = {TEST_TRIBE_TWO, TEST_TRIBE_ONE, TEST_TRIBE_ONE,
TEST_TRIBE_ONE, TEST_TRIBE_TWO, TEST_TRIBE_TWO};

        //Instantiate and add the 6 test contestants to the game.
        for (int i = 0; i < 6; i++){
            Contestant c = new Contestant(testID[i], testFirstName[i],
testLastName[i], testTribe[i]);
            g.addContestant(c);
        }

        //Start the season, so that contestants may be cast off, and the weeks
can be advanced.
        g.startSeason(TEST_BET_AMOUNT);


        // TEST 1: CAST OFF THE (TEST) CONTESTANT "JESSICA SIMPSON"
        g.getAllContestants().get(2).toCastOff();     //set the next contestant
to be cast off
        g.advanceWeek();                                          //advance the
week, committing the cast off

        // CONFIRMATION/OUTPUT: isCastOff returns true if contestant NOT casted
off
```
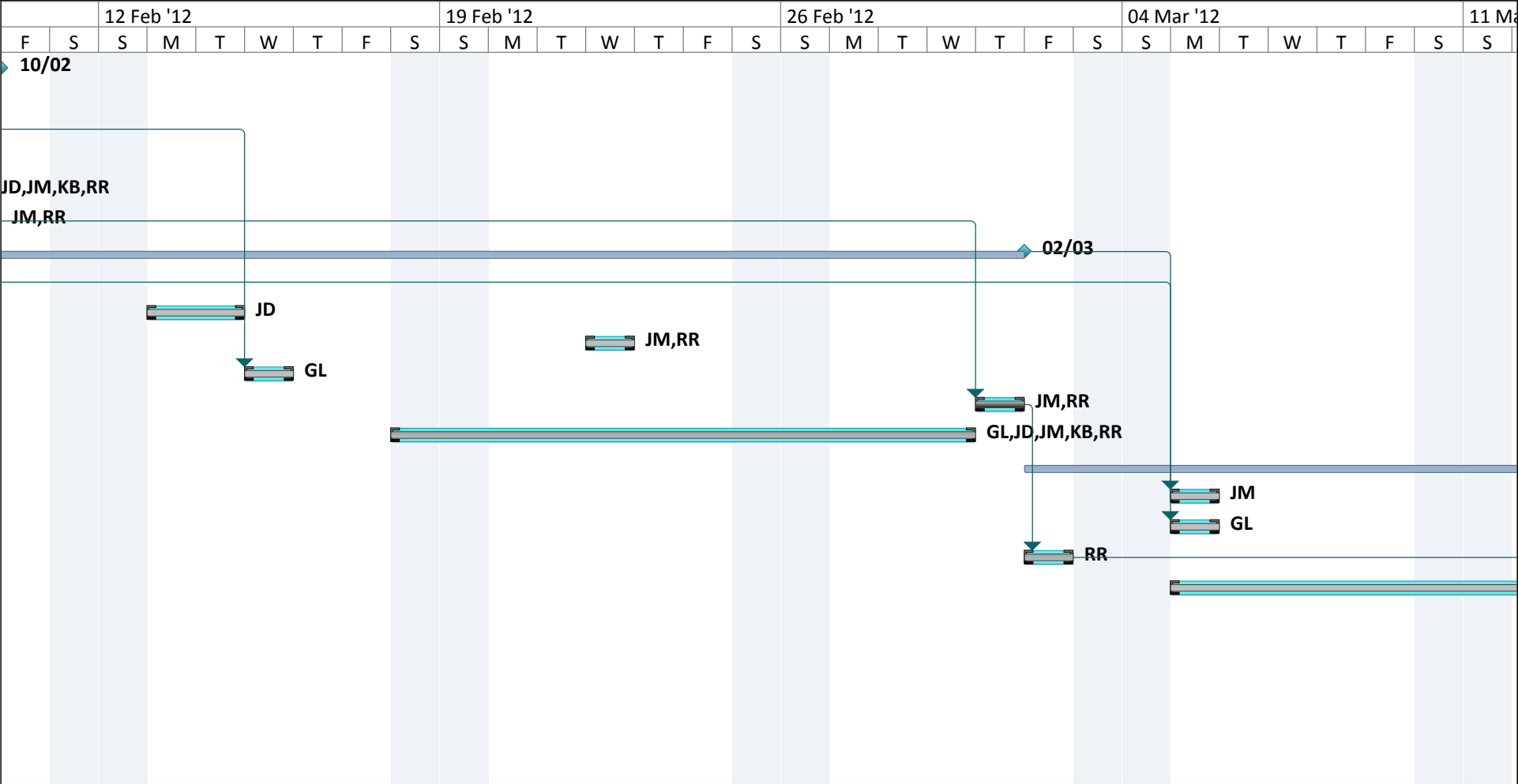
ID		Task Mode	Task Name	Duration	Start	Finish	29 Jan '12	05 Feb '12
1	✓	📌	**Milestone 1**	**0 days**	**Fri 10/02/12**	**Fri 10/02/12**		
2	✓	📌	Personnel Profiles	1 day	Sun 29/01/12	Mon 30/01/12	KB,JM,GL,JD,RR	
3	✓	📌	UML Class	1 day	Mon 06/02/12	Tue 07/02/12		KB
4	✓	📌	UML Use Case	1 day	Mon 06/02/12	Tue 07/02/12		JD
5	✓	📌	Learn MS project	3 days	Mon 06/02/12	Wed 08/02/12		GL,JD,
6	✓	📌	Project Plan	1 day	Thu 09/02/12	Thu 09/02/12		
7	✓	📌	**Milestone 2**	**0 days**	**Fri 02/03/12**	**Fri 02/03/12**		
8	✓	📌	Revised UML Class	1 day	Wed 08/02/12	Wed 08/02/12		KB
9	✓	📌	User Interface Design	2 days	Mon 13/02/12	Tue 14/02/12		
10	✓	📌	UML Sequence	1 day	Wed 22/02/12	Wed 22/02/12		
11	✓	📌	Class descriptions	1 day	Wed 15/02/12	Wed 15/02/12		
12	✓	📌	Project Plan 2	1 day	Thu 01/03/12	Thu 01/03/12		
13	✓	📌	25% Code	9 days	Sat 18/02/12	Wed 29/02/12		
14	✓	📌	**Milestone 3**	**0 days**	**Fri 16/03/12**	**Fri 16/03/12**		
15	✓	📌	Revised UML Class	1 day	Mon 05/03/12	Mon 05/03/12		
16	✓	📌	Testing Information	1 day	Mon 05/03/12	Mon 05/03/12		
17	✓	📌	Project Plan 3	1 day	Fri 02/03/12	Fri 02/03/12		
18	✓	📌	50% Code	9 days	Mon 05/03/12	Thu 15/03/12		
19		📌	**Milestone 4**	**0 days**	**Mon 09/04/12**	**Mon 09/04/12**		
20		📌	User Manual	2 days	Thu 05/04/12	Fri 06/04/12		
21		📌	100% Code	14 days	Mon 19/03/12	Thu 05/04/12		
22		📌	Project Plan	1 day	Sat 07/04/12	Sat 07/04/12		
23		📌	Presentation preparation	1 day	Sun 08/04/12	Sun 08/04/12		
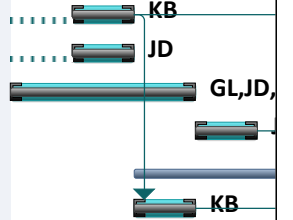
Project: 2212
Date: Fri 16/03/12

Task		External Milestone	◆	Manual Summary Rollup
Split	··················	Inactive Task		Manual Summary
Milestone	◆	Inactive Milestone	◇	Start-only
Summary		Inactive Summary		Finish-only
Project Summary		Manual Task		Deadline ⬇
External Tasks		Duration-only		Progress

ar '12							18 Mar '12							25 Mar '12							01 Apr '12							08 Apr '12		
M	T	W	T	F	S	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W

16/03

GL,JD,JM,KB,RR

09/04

Task		External Milestone	◆	Manual Summary Rollup
Split	....	Inactive Task		Manual Summary
Milestone	◆	Inactive Milestone	◇	Start-only	⌐
Summary		Inactive Summary		Finish-only	⌐
Project Summary		Manual Task		Deadline	⬇
External Tasks		Duration-only		Progress	

Project: 2212
Date: Fri 16/03/12