






# My Report

Last Modified: 08/25/2015  
Completion Status: Completed





## 1. 1. What is your current position?

#	Answer	Bar	Response	%
1	Undergraduate Student		2	13%
2	Graduate Student		9	60%
3	Developer		1	7%
4	Faculty		1	7%
5	Other		2	13%
	Total		15	

Other
Postdoc
Postdoc


Statistic	Value
Min Value	1
Max Value	5
Mean	2.47
Variance	1.55
Standard Deviation	1.25
Total Responses	15

2. 2. What is your year of experience on Java programming?

#	Answer	Bar	Response	%
1	Less than a year		2	13%
2	1 - 2 years		2	13%
3	2 - 5 years		4	27%
4	more than 5 years		7	47%
	Total		15	

Statistic	Value
Min Value	1
Max Value	4
Mean	3.07
Variance	1.21
Standard Deviation	1.10
Total Responses	15

3. Please make sure that you have downloaded all the requirement files since you cannot go back to this page.

#	Answer	Bar	Response	%
1	Yes		15	100%
	Total		15	

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	15

4. Q1.1 Please write a database related summary for the method addTimeSlot in Class com.umass.code.Timeslots

Text Response	
create one record with the start time, end time and the type of the slot and then add it to the table timeslots if it is not in it	
The method is used to insert the a new time slot. There are two types of time slot. The function insert the time slot into database if the parameters are valid.	
Add new time slot without duplicate time slots.	
This method, given (i) the start hour, (ii) the end hour, and (iii) a type, allows to add a new time slot, if it is not already present in the database. @pre: if the hours are not correct or the type is invalid, a IllegalArgumentException is thrown.	
check if all entries in the database has time slots, if not write the time slot	
- finds existing timeslots between start and end hour - if none, new timeslot is added in transaction for specified data - DBAnnotation.annoate() seems to be a dummy method doing nothing	
This method allows to save a time slot into database. The input values are start time, end time and the slot type. The valid slot time types are 1 and 2. The return value of this method is a boolean value according the result of the operation.	
The method is a static method that adds a time slot entry to the database (table 'timeslot'). Parameters of the method are the start and end hour of the time slot as well as its type. First, the method checks for illegal arguments and throws an exception if hours are not plausible (as specified in method 'areHoursCorrect()') or do not fit to the type of slot (see 'isTypeCorrect()'). Then, the database connection is retrieved and the database operation is prepared: (i) it is checked that not already a time slot with the same start and end hour exists in the database (otherwise, an exception is thrown), (ii) an insert statement for the timeslot is assembled and executed. A non-existing database connection results in no reaction; any SQL-related exceptions are caught and trigger an error message to be printed. The method always returns 'true' (the local variable 'isAdded' is never set to 'false').	
begin transaction; create table timeslots ( StartHour integer not null, EndHour integer not null, TimeslotType integer, CONSTRAINT start_End_Hours UNIQUE (StartHour, EndHour) ); INSERT INTO timeslots (StartHour, EndHour, TimeslotType) Values (startHour,endHour,type); commit;	
The starting hour and end hour are checked against the time slots currently in the database. If a time slot with the same start and end hour exists, then the new time slot is not created. Otherwise it is.	
First, it will check if the selected time slot does exist in the database, if so then it will throw an exception to the user with a message 'Already existing time slot'. If not, it will insert a record in the database that contains the startHour, endHour and the type of the time slot.	
The first sql query in this method retrieves the all of the available timeslots for the database and checks if a time slot to be added already exists. If the timeslot does not exist, it inserts a new TimeSlot into the table. There are constraints on the second database query for the Database columns "StartHour", "EndHour", and "TimeSlotType", these must all be of type integer and the "endHour" must come chronologically after the "startHour" and the timeSlotType must be a predefined integer.	
Add a new time slot to the "timeslots" table if it does not already exist, otherwise an exception is raised. Since the CourseSchedule uses these time slots to schedule things, it would be necessary to review that class too for any TimeSlot's method modification.	
Retrieve from the database a timeslot object (consisting of timeSlotID, startHour, endHour, timeSlotType) with the particular timeSlotID that is given as parameter. throws IllegalArgumentException if timeSlotID is not valid.	
//retrieve all time slots and check if the time slot already exists Select * FROM timeslots WHERE starthour = ? and endhour = ? //if all the pre-conditions are satisfied the new time slot can be added INSERT INTO timeslots (StartHour, EndHour, TimeslotType) Values(?,?,?); // database annotations	
Statistic	Value
Total Responses	15

5. Q1.2 Please write a database related summary for the method addSchedule in Class com.umass.code.CourseSchedule

**Text Response**

add a new record to the table courseschedule with the IDs of the offer, time slot and classroom

This function is called to insert the schedule information about a class.

Schedule the course when an empty slot is found.

Given (i) the offer ID, (ii) the classroom ID, and (iii) the timeslot ID, the method adds a new schedule to the table Course Schedule.

add course information into database

- inserting new courseschedule using provided arguments - DBAnnotation.annoate() seems to be a dummy method doing nothing

Allows to schedule the course offering with the offer, class and time slot id.

The method is a static method that adds a schedule entry to the database (table 'courseschedule'). Parameters of the method are the IDs of the offer, the classroom and the time slot. First, the database connection is retrieved. Then, the database is updated by assembling and executing an insert operation on the respective table. A non-existing database connection results in no reaction; any SQL-related exceptions are caught and trigger an error message to be printed. The method returns 'true' if the database update was executed and 'false' otherwise.

begin transaction; create table courseschedule ( OfferID integer not null, ClassroomID integer not null, TimeslotID integer, ); Insert into courseschedule (OfferID, TimeSlotID, ClassroomID) Values(offID,timeID,classID); commit;

This method adds a new Course to the database, however it does not perform checks on the classroom and time. This method is meant to be called only after other methods have been called that check if this course already exists. Thus, this method only adds the course into the courseschedule table

First thing first, it will try to establish a connection to the database. Get offID, classID and timeID from the database. Insert the passed parameters after they are checked from the database. Insert the values into table 'courseschedule'. Then use preparedStatement to update the inserted data. Then print: Adding course schedule ..."

There is only one database query in this method and it inserts the OfferID, TimeSlotID, and ClassroomID, into the CourseSchedule table. There are constraints on each of these IDs in that they most likely must be unique when compared to existing entries in the database.

Add a new course schedule to the "courseschedule" table, where it requires an already created time slot. Modifications to CourseOffered would be required for any modifications on that method.

Inserts into the "courseschedule" table a new course schedule consisting of the offerID, classroomID and timeslotID. These values are given as parameters. These values will act as the primary key for the table and they must be valid, namely: - offerID should have a corresponding foreign key reference in "coursesoffered" table - classroomID should have a corresponding foreign key reference in "classroom" table - timeslotID should have a corresponding foreign key reference in "timeslots" table Returns true if the course schedule was inserted into the "courseschedule" table successfully, and false otherwise.

database annotations Insert into courseschedule (OfferID, TimeSlotID, ClassroomID) Values(?,?,?)

Statistic	Value
Total Responses	15

6. Q1.3 Please write a database related summary for the method emailFirstStudentOnWaitList in Class com.umas.code.WaitList

**Text Response**

get student info based on the offerID, send email to this student and insert a new record to the table emailedwaitlist with the ID of the student, offer and the timestamp

It adds the first student of the wait list to email list, and remove the record on the wait list. At last, it sends email to the student.

Email new student for the same offer id.

Given the offer ID, this method allows to add the student to emailed list.

add student information and time to the database

- get all students registered already for course to check whether still space - find index of first person on waiting list for the course under investigation - if waiting list not empty, return this person (i.e., student) - insert that student in emailed waiting list, and remove person from waiting list - transaction committed only when actual email worked

Allows to search for the first student on a wait list. If the student is found, a log is saved into the emailedwaitlist table.

The method is a static method that triggers an email to a student who is first in the waiting list for a course offer. The only parameter of the method is the ID of the course offer. Before actually triggering the email, the methods checks whether the course can accommodate new students by looking up in the database whether the course with the respective ID still has enough capacity (see 'canCourseAccomodateNewStudentFromWaitList()'). If not or if the course does not exist, the methods stops. Otherwise, the first student on the waiting list is retrieved from the database (see 'getStudentFirstOnWaitList()'). If this student exists (not 'null'), the method proceeds with retrieving the database connection and executing a insert operation on the log of emailed waiting list (table 'emailedwaitlist'). Then, the student is removed from the waiting list by updating the respective table (table 'waitlist', see 'removeFromWaitList()'). Finally, an email is assembled and sent to notify the student about the possibility to register for the course.

begin transaction; create table emailedwaitlist ( StudentUIN integer not null, OfferID integer not null, TimeEmailed datetime, ); INSERT INTO emailedwaitlist (StudentUIN, OfferID, TimeEmailed) Values(studentUIN,offerID,timeStamp); commit;

First, a check is performed to ensure that the class has sufficient capacity to accomodate this student in addition to the currently registered students in the class as well those who have already been emailed from the wait list. Only if these conditions are satisfied, then the student is added to the emailed list and an invitation email for registration is sent out.

Try to establish a connection to the database. Insert StudentUIN, OfferID and TimeEmailed into table 'emailedwaitlist' using preparedStatement. If there is an SQLException, print it.

The database related code in this method inserts a student that is retrieved from the waitlist and adds them to a list of students who are on the wait list, but have been emailed. The constraints for the the StudentUIN, OfferID, and TimeEmailed, which must fit into the db structure.

Get a student from the "waitlist" table, then add it to the "emailedwaitlist" table while removing it from the "waitlist" table. Since this method is not used by any other one, any modification on it would not require external verification.

For a given offerID course, the method checks if there is enough capacity in the offerID course. If there is no capacity, no error is returned, no flags are set. If there is capacity in the course, the following operations are performed: - the first student from the course wait list queue is removed - this student is added to the "emailedwaitlist" table - the student will receive an email in which he will be informed that they have a 12 hour window to register for the class The offerID has to be valid.

// add student to emailed list INSERT INTO emailedwaitlist (StudentUIN, OfferID, TimeEmailed) Values(?,?,?) database annotations //remove the student from wait list  
removeFromWaitList(student, offerID)

Statistic	Value
Total Responses	15

7. Q1.4 Please write a database related summary for the method initializeAllComboBoxesInRegistrationTab in Class com.umas.frontend.StudentUI

**Text Response**

build the department, course and offer model based on the user's input

It reads the department courses and offerings information and fill them in the combobox.

Initialize student's course registration.

This method loads all the departments present in the Database

show available courses from data base

- read all departments from DB - eager loading of first department's courses from DB, as well as of the first course's offerings - current student's eligibility for default offering is checked - number of people on emailed waiting list is checked as well for space calculations - also a check if the student is already registered or on waiting list, or if the waiting list is empty

This method is responsible for load the data of all combo boxes in the registration tab of the student user interface.

The method is a private method that initializes combo boxes in the students' registration tab for courses. To this end, it retrieves all departments from the database and uses these to a first combo box. The second combo box is filled with with courses offered by the first department of the list, which is also retrieved from the database. The course IDs also included in this database request are added to a third combo box. Based on information about the student and selected course from the database, buttons to register and to join the waiting list are activated.

I don't know.

This method populates the check boxes for Registration and is dependent on database queries to fetch departments and the courses associated with each department. In order to get all courses, a join is performed with the courses table.

Get all departments and courses related to each department. As well as, the status of each course (Registerable, WaitListEligible).

There are few callees that contain database realted code. the getDepartmentCourses() method returns all of the courseID's for a given DepartmentID, the constraint here is that the DepartmentID must be a valid ID that currently exists in the database. The getCurrentOfferings() method returns the current course offerings for a particaulr courseID. The constraint here is that the CourseID must be valid.

That method gets informations from different tables such as the ones for the departments, courses, students, etc. This method is not used to modify things in the different tables, it is only used to update combo boxes GUI interface.

Retrieves from the database all departments, all courses offered by the departments, and all course offerings corresponding to the department courses. This information is populated into combo boxes. This method also provides the student a button to register for a class, or a button to be added to the wait list of a class.

"Select \* FROM department" //if size >0 Select CourseID FROM department natural join courses WHERE DepartmentID= ? //if size >0 Select \* FROM coursesoffered natural join courseschedule WHERE coursesoffered.CourseID= ?

Statistic	Value
Total Responses	15

8. Q1.5 Please write a database related summary for the method  
actionPerformed in Class com.umas.frontend.StudentUIDetails

**Text Response**

update student info in database with new name, show the result messages with different dialog.

It is the event handler which update the existing student's name in database.

When the button "btnUpdateName" is clicked, actionPerformed() is called to update the student's name if the person already exists in the database. Then, it initializes the view details including the student's name, UIN, user name, department, present name, present user name, and GPA and clear the new name after.

There are no options related to the management of the database.

update information in the database

NOTE: There are multiple methods named "actionPerformed" in StudentUIDetails.java, I picked the first one on line 236. - name of student is updated in database with new name

This method is invoked when: - The update name button is selected and allows to update the student name in the database. - The update user name is selected and allows to update the student user name. - The change password button is selected and allows to change the student password. - The login buttons is select. - The add and update applications details is selected. This actions allows to save information related to work experience, a list of student' skills and the scaled score.

[Sorry, could not write a summary because there exist multiple methods called 'actionPerformed' in the class.]

I don't know.

This method tries to update the students name by updating the information in the Login table. The input is checked for null and empty conditions.

Updates student's name based on UIN.

There are two nested callees to get any database code here first the updatePeople() method and then the UpdateNameIntoPeopleTable() method that sets a persons name in the People Table according to their UIN, therefore the constraint is the UIN must be valid according to the database structure.

I did not manage to find such method.

(There are 6 actionPerformed methods in com.umas.frontend.StudentUIDetails so I provided a brief description for all) These actionPerformed methods perform the following operations in the database: - Update the student name - Update the student user name - Update student password - Checks the username and password of a student and authenticates - Inserts into the "applicationdetails" table a new student job application - Updates the student job application details

SELECT Username, Password FROM logindetails WHERE username = ? AND Password = ? database annotations




Statistic	Value
Total Responses	15



9. Q1.6 Please write a database related summary for the method populateAdmins in Class com.umas.code.PopulateData




Text Response	
Randomly select 50 records from table name3 and try to add them as admins.	
It randomly select 50 administrators from all departments.	
Populate database with 50 administrators.	
Firstly, this method reads 50 random departments from the database. Then, 50 new admins are added.	
add admins to database	
- get list of all departments - 50 random semester ids are generated - to generate random admins, the system needs to check for each admin added: - whether the name already exists in the Username table - same, but for the logindetails table - then, if not yet existing, the new (fictional) data is inserted into logindetails - again a check if the name already exists, this time in the People table - the name is added in the People table and the primary key is requested - whether the person exists in the employee table - if not, the person is added in that table	
Search for 50 names in the names3 table and for each one get a random department and save the information into the login and people tables.	
The method is a public method that adds 50 random admins to the database and returns 'true' if successful. To this end, it first selects 50 random names from a names table in the database. For each of these names, a random department from those existing in the database is selected and the name is added as a new admin to the database.	
I don't know.	
This method adds Administrators and it is dependent on the departments table as well as the semester. It fetches departments from the departments table and then gets names from those departments to add as Admins.	
Connect to DB. Select 50 administrators from table 'names3'. The random number will be between 0 and the total number of names in each department. Add admin to each department. Great use of sleep :)	
This code has one database query directly in the method that extracts all of the entires from the names3 table in random order. There are two callee methods that contain Database code: getAllDepartments() and addAdmin(). The constratins for these are none for the getAllDepartments() method and for the addAdmin() method there are two nested callees that add the Admin to teh database and they have constraints of the name of the person and department number being valid in the databse strcuture.	
Populate administration by selecting a maximal of 50 random elements from the "names3" table. This method is used for populating, thus any modification on the method may have an impact on that process.	
Fifty randomly selected employees are added into the database as administrators.	
Select * FROM department Select * FROM names3 order by rand() LIMIT 50	
Statistic	Value
Total Responses	15

10. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		10	67%
2	The description misses some important information to understand the database-related operations		4	27%
3	The description misses the majority of the important information to understand the database-related operations		1	7%
Total			15	



Statistic	Value
Min Value	1
Max Value	3
Mean	1.40
Variance	0.40
Standard Deviation	0.63
Total Responses	15

11. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		7	47%
2	The description contains some redundant/useless information.		7	47%
3	The description contains a lot of redundant/useless information.		1	7%
	Total		15	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.60
Variance	0.40
Standard Deviation	0.63
Total Responses	15

12. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?



#	Answer	Bar	Response	%
1	The description is easy to read and understand.		13	87%
2	The description is somewhat readable and understandable.		2	13%
3	The description is hard to read and understand.		0	0%
	Total		15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.13
Variance	0.12
Standard Deviation	0.35
Total Responses	15

13. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)




Text Response	
my answer is easy to read and understand, but maybe a little short, lacking some info	
They are correct.	
Relevant information related to the method has been provided.	
more accurate than manual checking	
Information overload for the varchar limits and referential constraints, because too much detail and in the end these kinds of errors are less likely to occur. On the other hand, the short snippets about null values were more useful, and so was the summary of operations.	
The information in the summary can be very large related with the foreign keys and the referential integrity. This summary shows the most important information related with database operations.	
Q1) The description does not make clear that the timeslot is not always added to the table. For instance, if the timeslot already exists, an exception is thrown. The relationship between 'queries' and 'inserts' does not become clear (but that would be very hard to achieve with an automatic approach).	
the database details which are specified are very useful and would be time consuming to look at manually (ie char constraints, etc) however, in addition to this information I think the comment needs to include the high level description. The fact that the DB is queried against the start/end hour and if this combination does not exist only then it is added.	
In the section "constraints": The sub-section "When inserting into table timeslots, make sure the referential integrity imposed by attribute(s) ..." is confusing. Maybe by using colours to describe the relation it would be less confusing. The same with the documentation of the foreign keys. UML diagrams would make it easier for anyone to comprehend the referential integrity of the DB.	
This summary misses the potential dependency of the EndTime needing to come after the start time chronologically. The sections about the referential integrity and adhering the varchar limits are little too long and detailed. I would like to have this information in the form of external documentation though.	
The summary made by the system was complete, precise, and easy to understand (even if sometimes there are many dependencies and links).	
The method also performs a select operation that is not described in the automated method summary The following points do not need to be displayed from the beginning, as they take a lot of space. Instead, they could be expanded if they user is interested in these details: - Make sure the strings to be stored in timeslots do not overflow the varchar limits: 2 (Grade, Level), 100 (FileLocation, FileName), 3 (PositionLevel), 20 (ExamName, Username), 45 (ClassroomName, Skillset5, Skillset3, Skillset4, Skillset1, Skillset2, Password, TaOfficeLocation, TaOfficeHours, CourseName, OfficeHours, ClassroomLocation, SemesterName, DepartmentName, PositionName, name, OfficeAddress, ReqSkillset4, Name, SkillsetName, ReqSkillset3, ReqSkillset5, names2, ReqSkillset2, names, ReqSkillset1, name1), - When inserting into table timeslots, make sure the referential integrity imposed by attribute(s) TaughtBy, StudentEnrollmentID, SemesterID, TaUIN, TimeSlotID, PostedByUIN, OfferID, Grade, CourseID, Username, ClassroomID, ApplicantUIN, JobInDepartment, StudentUIN, UIN, DepartmentID, PositionID is accomplished. The foreign keys in the table are the following: UIN people.UIN, JobInDepartment department.DepartmentID, DepartmentID department.DepartmentID, SemesterID semester.SemesterID, CourseID courses.CourseID, Username logindetails.Username, PostedByUIN people.UIN, OfferID coursesoffered.OfferID, ApplicantUIN people.UIN, TaUIN student.UIN, PositionID position.PositionID, StudentUIN student.UIN, TimeSlotID timeslots.TimeSlotID, ClassroomID classroom.ClassroomID, StudentEnrollmentID studentenrollment.EnrollmentID, Grade gradingsystem.Grade, TaughtBy people.UIN	
all descriptions given are useful and easy to understand	
Statistic	Value
Total Responses	13

14. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		14	93%
2	The description misses some important information to understand the database-related operations		1	7%
3	The description misses the majority of the important information to understand the database-related operations		0	0%
Total			15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.07
Variance	0.07
Standard Deviation	0.26
Total Responses	15

15. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		9	60%
2	The description contains some redundant/useless information.		5	33%
3	The description contains a lot of redundant/useless information.		1	7%
	Total		15	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.47
Variance	0.41
Standard Deviation	0.64
Total Responses	15

16. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		11	73%
2	The description is somewhat readable and understandable.		4	27%
3	The description is hard to read and understand.		0	0%
	Total		15	



Statistic	Value
Min Value	1
Max Value	2
Mean	1.27
Variance	0.21
Standard Deviation	0.46
Total Responses	15



17. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)



Text Response	
my answer is easy and short. My I think it is enough.	
The comments well explains the function.	
See comments on previous one.	
This summary describe the important action that is insert into courseschedule table but shows many details about constraints, data types and so on.	
Q3) I don't understand what "Insertions in table courseschedule do not require to set the attribute ClassroomID because it is auto-numeric" and related statements mean in this contexts. It seems contradicting to the rest of the summary (and the code).	
Same as previous answer, the database information is concise and on point, BUT, it misses the high level logical flow of the method. For example, it does not convey the message that this method is meant to be called after checking classroom and type. So from a database point of view the auto generated comment is good, but from overall point of view, it misses some required information.	
Same as 1.	
Again the referential integrity and varchar limits are cumbersome to read.	
Same as 1)	
description is usefull and easy to understand	
Statistic	Value
Total Responses	10

18. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		13	87%
2	The description misses some important information to understand the database-related operations		2	13%
3	The description misses the majority of the important information to understand the database-related operations		0	0%
Total			15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.13
Variance	0.12
Standard Deviation	0.35
Total Responses	15

19. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		10	67%
2	The description contains some redundant/useless information.		5	33%
3	The description contains a lot of redundant/useless information.		0	0%
	Total		15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.33
Variance	0.24
Standard Deviation	0.49
Total Responses	15

20. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		11	73%
2	The description is somewhat readable and understandable.		4	27%
3	The description is hard to read and understand.		0	0%
Total			15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.27
Variance	0.21
Standard Deviation	0.46
Total Responses	15

21. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

**Text Response**

My answer is easy, may miss some info like you need to check for some attributes, your info can not be NULL.

The comments well explain the function.

The operations, both top-level and delegated, are easy to read. The constraints are too detailed, some kind of prioritization would be needed.

The information about the db-related operations is very usefull and completeness.

Q1) Again, the conditions and relationships between the queries do not become clear (I'm not sure about the sequence of the queries - is it the sequence like in the code?). It would be also nice to see which columns are actually used from the queries. Q3) Several ways of querying table waitlist produce some redundant text - you have to compare one-by-one to find that nearly which points refer to the same table.

Same rationale as for the other questions. The auto generated DB comments are useful but only in addition to a high level explanation of the comment.

Same as 1.

I like how it has the references to the caller callee information, as this could help speedup development or maintenance tasks due to the ease of knowledge presented.



same as 1)

Very nice explanation of the chain call (for example): - It queries the table(s) emailedwaitlist via the chain-call  
com.umass.code.WaitList.canCourseAccomodateNewStudentFromWaitList com.umass.code.WaitList.getStudentsOnEmailList

description is usefull and easy to understand



Statistic	Value
Total Responses	11

22. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		12	80%
2	The description misses some important information to understand the database-related operations		3	20%
3	The description misses the majority of the important information to understand the database-related operations		0	0%
Total			15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.20
Variance	0.17
Standard Deviation	0.41
Total Responses	15

23. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		9	60%
2	The description contains some redundant/useless information.		6	40%
3	The description contains a lot of redundant/useless information.		0	0%
	Total		15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.40
Variance	0.26
Standard Deviation	0.51
Total Responses	15

24. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		9	60%
2	The description is somewhat readable and understandable.		4	27%
3	The description is hard to read and understand.		2	13%
	Total		15	

Statistic	Value
Min Value	1
Max Value	3
Mean	1.53
Variance	0.55
Standard Deviation	0.74
Total Responses	15





25. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

Text Response
My answer doesn't take the methods called inside this function into account. It lacks info about how the database are used. Well, I think that should be the summary for that function.
The comments are easy for read, but there are some redundant explanation.
Everything is there, but there is no hierarchical overview, it's basically a long, flat list of operations. Nesting of operations would be cool.
In this case repeat the statement related to the queries performed to the emailedwaitlist, studentenrollment. I think this is not necessary.
Q1) In this method, the result is not a change in the database but in the UI. To understand the method, the information which UI elements change is more important here than which database tables are queried. Q3) The list of queries become too long and the summary does not abstract, e.g., by summarizing all queries related to the same table.
Same rationale as for the other questions. The auto generated DB comments are useful but only in addition to a high level explanation of the comment.
Same as 1.
This comment is very useful, but all of the information presented is a little overwhelming. Again this would be useful as external documentation.
The summary provides all needed informations about the queries made to the different tables.
description is usefull and easy to understand



Statistic	Value
Total Responses	10

26. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		12	80%
2	The description misses some important information to understand the database-related operations		3	20%
3	The description misses the majority of the important information to understand the database-related operations		0	0%
Total			15	



Statistic	Value
Min Value	1
Max Value	2
Mean	1.20
Variance	0.17
Standard Deviation	0.41
Total Responses	15

27. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		7	47%
2	The description contains some redundant/useless information.		8	53%
3	The description contains a lot of redundant/useless information.		0	0%
Total			15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.53
Variance	0.27
Standard Deviation	0.52
Total Responses	15

28. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		8	53%
2	The description is somewhat readable and understandable.		7	47%
3	The description is hard to read and understand.		0	0%
	Total		15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.47
Variance	0.27
Standard Deviation	0.52
Total Responses	15

29. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

**Text Response**

Similarly, my answer lacks database info used by the functions called inside this one.

The same as last one, it contains some redundant information.

Similar to previous one, some more structure would be nice.

Statements related to queries to the same table would be summarized.

[Sorry, there are multiple methods with this name (see above) - please ignore my answer for this question]

Same rationale as for the other questions. The auto generated DB comments are useful but only in addition to a high level explanation of the comment.

Same as 1.



Sadly, I could not find the method, so I provided answers like the previous ones (the tool were looking really good so far).

I just wanted to mention that there is no "com.umas.frontend.StudentUIDetails.actionPerformed" instead there are 6 methods with the name actionPerformed that are hosted inside anonymous classes inside the StudentUIDetails(Student s) constructor.

description is usefull and easy to understand



Statistic	Value
Total Responses	10

30. Q1. Only focusing on the content of the description without considering the way it has been presented, do you think the message is complete?

#	Answer	Bar	Response	%
1	The description does not miss any important information.		10	67%
2	The description misses some important information to understand the database-related operations		5	33%
3	The description misses the majority of the important information to understand the database-related operations		0	0%
Total			15	



Statistic	Value
Min Value	1
Max Value	2
Mean	1.33
Variance	0.24
Standard Deviation	0.49
Total Responses	15

31. Q2. Only focusing on the content of the description without considering the way it has been presented, do you think the message is concise?

#	Answer	Bar	Response	%
1	The description contains no redundant/useless information.		10	67%
2	The description contains some redundant/useless information.		5	33%
3	The description contains a lot of redundant/useless information.		0	0%
Total			15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.33
Variance	0.24
Standard Deviation	0.49
Total Responses	15

32. Q3. Only focusing on the content of the description without considering the completeness and conciseness, do you think the description is expressive?

#	Answer	Bar	Response	%
1	The description is easy to read and understand.		12	80%
2	The description is somewhat readable and understandable.		3	20%
3	The description is hard to read and understand.		0	0%
	Total		15	

Statistic	Value
Min Value	1
Max Value	2
Mean	1.20
Variance	0.17
Standard Deviation	0.41
Total Responses	15



33. Q4: Please provide the rationale for your answers to Q1 to Q3. (Open question)

**Text Response**

My answer only mentions about queries from the database, lacking info about the inserting stuff. My answer is relatively easy.

It provide critical and correct explanation.

Delegated accesses to tables People and logindetails are missing.

The important related-db action is the insertion into the employees table and the summary contains this information.

Q1) relationship between queries not clear (see above) [Link from this page to the summary incorrect, jumpst to the top, had to search for the correct method - no problem, just FYI]

Same rationale as for the other questions. The auto generated DB comments are useful but only in addition to a high level explanation of the comment.

Same as 1.

It does not tell the limit of 50 elements.

description is usefull and easy to understand

Statistic	Value
Total Responses	9

34. General question 1: Is our generated summary useful for understanding the database usages in the system?

#	Answer	Bar	Response	%
1	Yes. It's useful. Why?		15	100%
2	No. It's not useful. Why?		0	0%
	Total		15	

Yes. It's useful. Why?	No. It's not useful. Why?
explains how the data are queried from and inserted into the database. How the database are used through the call path of a function.	-
It lists the purposes of the functions	-
Yes. Definently. The reason is that the automatically generated summaries contain, in general, all the relevant information needed to understand the database usages even though I find the summaries a bit long/detailed in some cases.	-
yes	-
Pointing out delegated accesses is very useful, would be even better if these would be nested/structured hierarchically. The constraints part is too detailed (very long lists of limits, ...), so ordering these based on priority and maybe hiding the least important ones until someone clicks would be better.	-
The generated summaries are useful because shows all related db operations and also shows another information related with the constraints, data types. With the constraints and validations allows to developer to understand any business logic restrictions.	-
It summarizes the database accesses efficiently that might be spread over many different methods. Even if all related database operations are contained in the respective method directly, the summary is much easier to read than finding the specific statements in the code.	-
Yes, but as mentioned before, this is useful only when the developer already understand the logic behind the method. Once the purpose and flow of the method are clear, then it is very useful to consider the DB restrictions on the different tables. Best example is probably the add new course to database, where the method assumes that previous checks have been performed on the input parameters.	-
Amazing. It makes it easier to find the relations created by method chaining.	-
The callee information, database actions being performed and the constraint information would all function great as reference materials for maintenance and evolution tasks.	-
Provided all the needed information to understand how the method interacts with the DB's tables	-
It's like an additional Javadoc that is specialized on database usage. It provides a quick insight to developers into the type of operations performed. The chain-calls are a very nice feature.	-
All descriptions given are useful	-

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	15

35. General question 2: What software engineering tasks would you use this type of summary for? (Open question)

**Text Response**

If you want to change the structure of a database, you can refer to this summary and change the statements accordingly.

understanding a project quickly

These automatically generated summaries may serve for redocumentation tasks and can be also investigated in the context of bug localisation (as queries) for the sake of improving its accuracy.

1) Impact Analysis; 2) Implementation of new functionalities based on such methods.

coding

Definitely for debugging and for profiling. Likely also when changing code that interacts with the database.

This information can be useful when: - I need add a new feature, I can understand the related db actions of any existing method. - To fix a bug. - To build the business process.

Understanding the code in general. I could also image that it is particularly helpful for debugging database-related errors (wrong updates, wrong implications drawn from the data) as well as performance problems due to unnecessary database queries.

I don't know

Code maintenance, addition of new features

No doubt, documentation.

As I said previously, I would use this for external documentation for reference when trying to understand database related source code.

For any task that needs a knowledge on how the system deal with the DB, such as: 1)Refactoring 2)Adding new features that need an interaction with the DB 3)QA analyses (e.g., to know if we have too many dependencies) 4)Find bottle necks where failure is not an option (e.g., tables holding information used by many methods)

Direct software programming and development of documentation

Statistic	Value
Total Responses	14