30

TENTAMEN (EXAMINATION)

Tentamensdatum/*Examination date:* 16-10-28
(åå-mm-dd/*yy-mm-dd*)

AID-nummer
AID number

Ifylles av student
Completed by student: 1 7 7 8

Ifylles av vakt
Completed by supervisor: 1 7 7 8

Kurskod/*Course code*: TDDC88 Provkod/*Exam code*: TEN1

Kursnamn/*Course title*: Programutvecklingsmetodik

Antal inlämnade blad/*Total number of sheets enclosed*: 13

Institution/*Department*: IDA

Anvisningar/*Instructions*

1. Skriv AID-nummer, datum, kurskod och provkod på varje blad som lämnas in/
   Write AID number, date, course code and exam code on every sheet that is handed in
2. På varje papper får högst en uppgift lösas om inget annat anges/
   Maximum one task per sheet unless otherwise instructed
3. Skriv endast på papprets ena sida om inget annat anges/
   Use only one side of each sheet unless otherwise instructed
4. Numrera de papper som lämnas in/*Number every sheet that is handed in*
5. Använd inte röd penna/*Do not use a red pen/pencil*

Sen inlämning ☐
Late hand in

Klockslag _____
Time

Orsak _____
Reason

Markera behandlade uppgifter med X/*Mark tasks attempted with an X*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Erhållna poäng *Points obtained*	X	X	X	X	X	X	X	X	X						
	9	8	8	8	8	9	19	9	8						
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Erhållna poäng *Points obtained*															

∑ Poäng/*Points:* 106 (20) Betyg/*Grade:* 5 Bäst på! Kg tentan.

Examinator/*Examiner:* _____

LINKÖPINGS UNIVERSITET

1

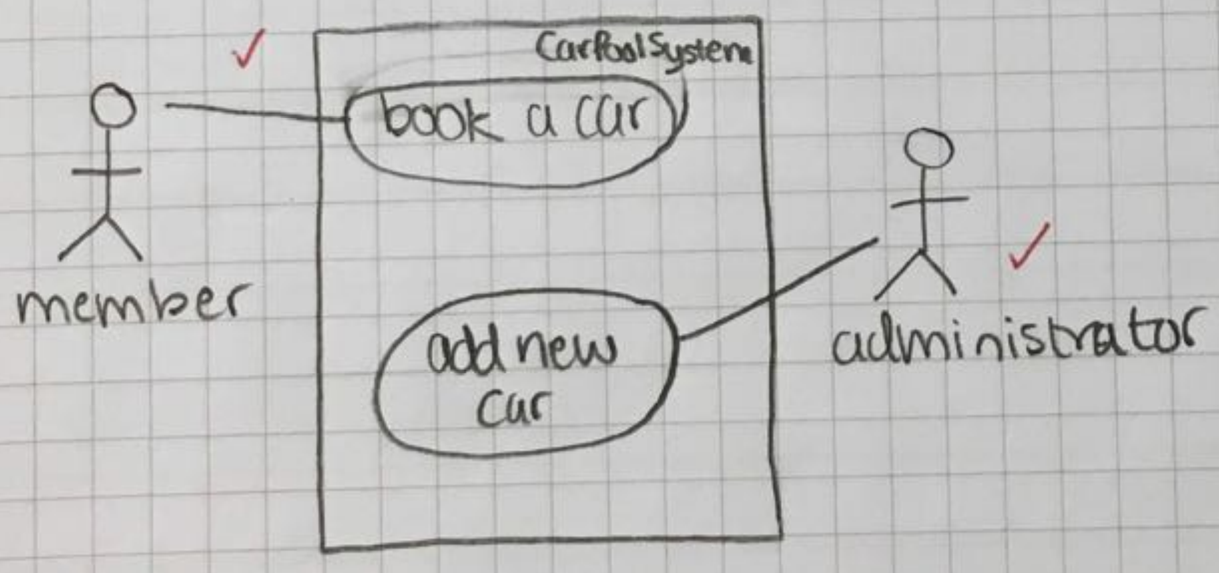
1) a) C ✓

b) Use case 1: A member book a car. ✓

The member fills in its username and password in the log-in form and press the "log-in" button. A message confirms that the log-in was succesful. The member press the tab "book a car". The "booking-page" is shown. The member fills in date, time, type of car and location for picking up the car in the form. The member press the "search for available car" button. A list of cars available is shown. The member selects one and press the "book" button. A message confirms the booking and the member is redirected to the "bookingpage".

Use case 2: An administrator adds a new car to the system.

The administrator logs in to its account by filling in the username and password in the login form. The administrator is logged in and then press the "add new car" button and is directed to a new page showing a form to fill in details about the car. The administrator fills in the form and press the "add" button. A message confirms that the car has been added and the start-page is loaded. 4



c) 2.1 Product perspectiv. Here a would put some information about the buisness case, like what benefits the company would get from the system. In 1b) this might be something about that the system would make it easier for user to book cars which will attract more users.

2.2 Product function. Here I would list some functions that the product should have. In 1b) I might add the use-cases.

2.3 User characteristics. Here I would put information about the users who should use the system.

LINKÖPINGS UNIVERSITET

AID-nummer: *AID-number:* 1778	Datum: *Date:* 16-10-28	Blad nummer: *Sheet number:* 2
Kurskod: *Course code:* TDDC88	Provkod: *Exam code:* TEN1	

1c) like what skills they have, so that the system can be adapted. In 1b) the description might be "general technical knowledge" ✓

2.4 General constraints. If there are any general constraints that needs to be considered. For example that the lock/unlock with mobile only works for iphones. ✓

4

LINKÖPINGS UNIVERSITET

AID-nummer: *AID-number:* 1778	Datum: *Date:* 16-10-28	Blad nummer: *Sheet number:* 3
Kurskod: *Course code:* TDDC88	Provkod: *Exam code:* TEN1	

2

(2) a) B, D

b) One stakeholder is the customer that has ordered this system. A stakeholder is anuone who has interest in the project / product and of course the customer has. why?

An other stakeholder is the developers that are going to develop the system and therefore also is a stakeholder. why?

3

c) One benefit with the waterfall method is that it is very well adapted to common project management "parts" like milestones and delivariables.

An other benefit is that it is simple and easy to understand.

One potentia problem is that it doesn't explicit include risk management which pushes the risks forward.

3

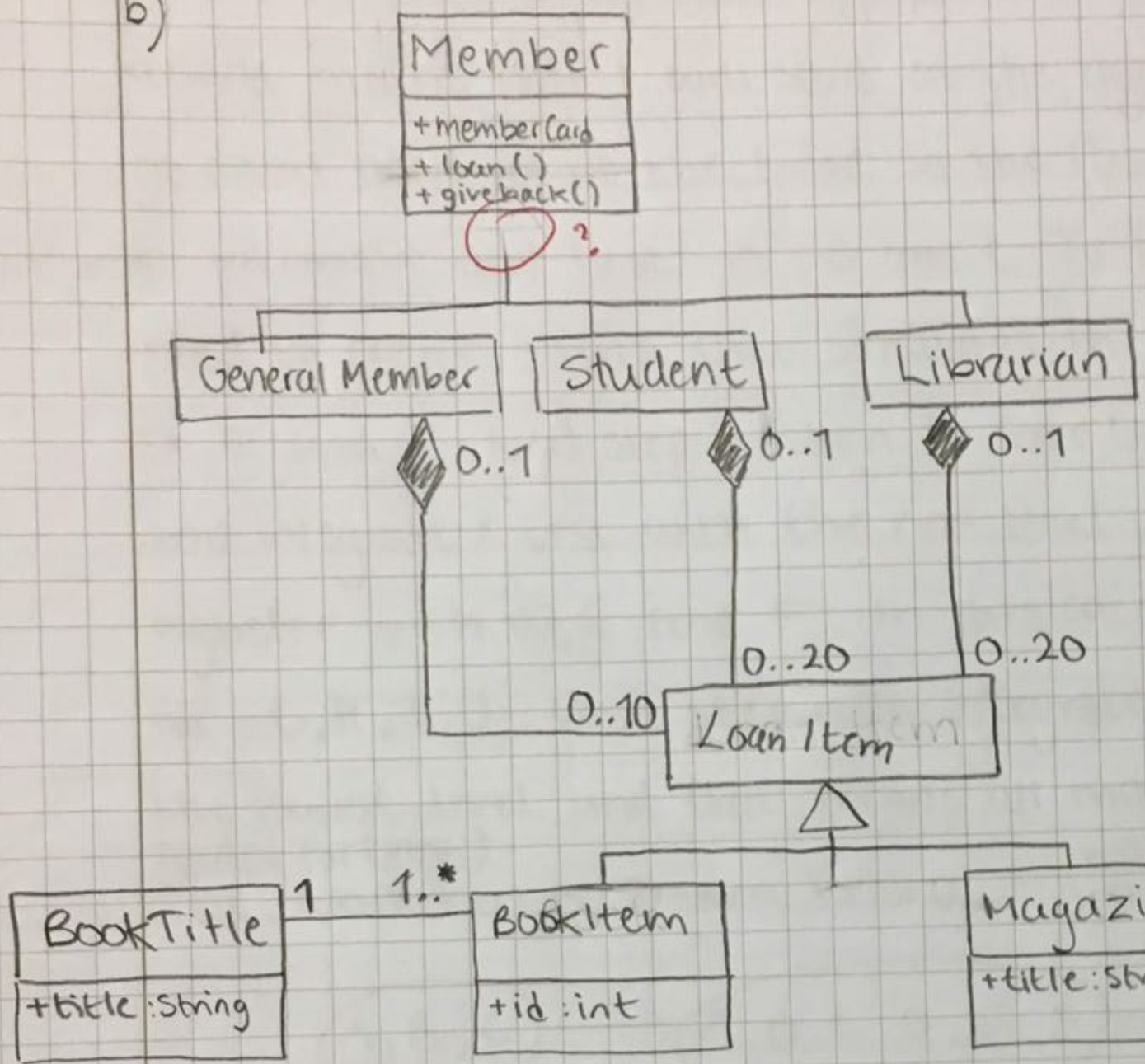
The difference between the waterfall method and the V-model is that each phase (like requirements, design, implementation etc) are only done once in the watefall, but in the V-model the process is iterative so the phases are happen again, which makes it easier to take feedback into consideration.

18

③ a) B, D

2p

b)



Kanns inte ett item om det inte lånas?

2p

c) I assume that the variant of the client-server architecture is a 2-tier thin-client, this means that there is a heavy load on the server and that there is a lot of network traffic between client and server. This can then leads to the long response time. One way of solving this is to make the architecture a two-tier thick-client instead which will reduce the load on the server and probably speed things up.
An other thing is to "scale up" so that the system gets higher capacity which will have a good impact to the response time.

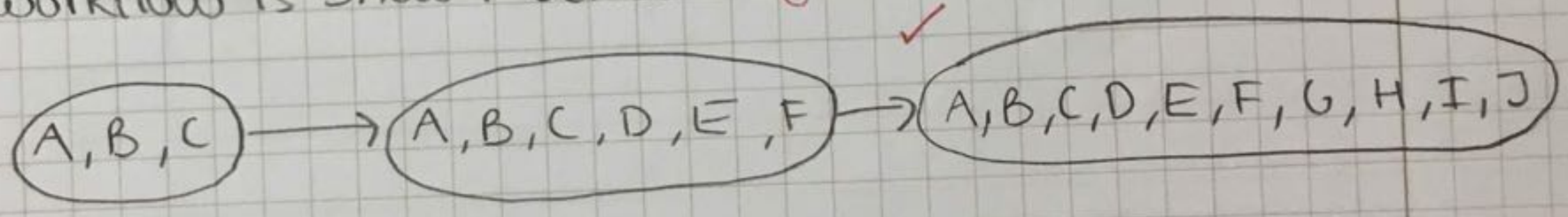
4p / 8p

LINKÖPINGS UNIVERSITET

AID-nummer: *AID-number:* 1778	Datum: *Date:* 16-10-28	Blad nummer: *Sheet number:* 5
Kurskod: *Course code:* TDDC88	Provkod: *Exam code:* TEN1	

(4) a) C

b) I will select a top-down approch for the testing & integration which means that I will start at the highest level and then go down one level at the time. So the first thing to do is to integrate and test A, B and C. To be able to do so I need to create stubs that simulates the input/output from D, E and F. Next step is then to take this integreted parts and integrate & test with the next level, which will be togheter with D, E and F. In this case stubs are needed for G, H, I, J. Last step is to integrate and test with the lowest level and that means all modules together. (no stubs needed for that).
The workflow is shown below very nice!



In total 7 stubs are needed. The top-down approch is better because it makes it possible to detect defects in the general design at an early state, and also it does not need drivers which often is harder to write than stubs.

c)
- Full path coverage means that all possible paths should be executed at
- Branch coverage means that all possible branches from a desicion points should be executed at least once.
- statement coverage means that every statement should be executed at least once.

se example on next page.

1

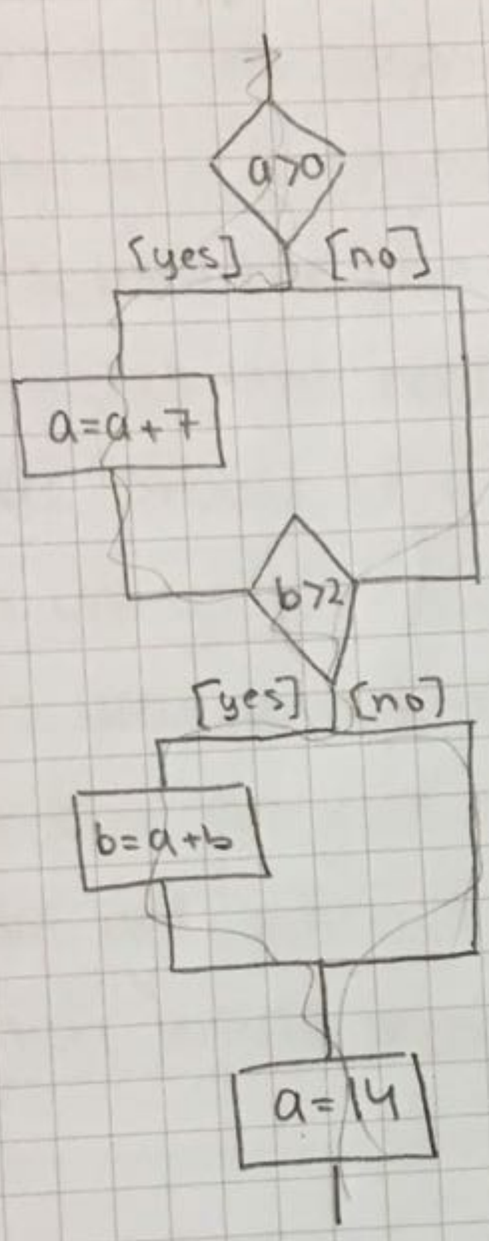
4

AID-nummer:

AID-nummer: *AID-number:* 1778	Datum: *Date:* 16-10-28	Blad nummer: *Sheet number:* 6
Kurskod: *Course code:* TDDC88	Provkod: *Exam code:* TEN1	

LINKÖPINGS UNIVERSITET

This flow graph satisfies whats required, because:



the minimal number of test-cases for full path coverage is 4, (left,left; right,right; left,right; right,left) which is strictly larger than the minimal number for branch coverage which is 2 (for example one test case that only left and one only right)

which is strictly larger than the number of test-cases for statement coverage which is 1. (you only need a test case that takes the left in both decision points)

4

(by "left" I mean that a>0 in the first desicion point and so on).

LINKÖPINGS UNIVERSITET

AID-nummer: *AID-number:* 1778	Datum: *Date:* 16-10-28	Blad nummer: *Sheet number:* 7
Kurskod: *Course code:* TDDC88	Provkod: *Exam code:* TEN1	

5. a) B, D

2

b) A suitible process area to work with can be project planning. The purpose of this area is to create and maintain a project plan that defines the project. The work with the plan involves doing estimations, define activities that needs to be done and by who. Also, there should be a commitment to the plan. The plan can change if needed. ++

This area would help us because if we have a well-done plan that everyone is commited to it would be clear what and when things should be done and by who + which would improve the efficiency.

3

from when?

c) The time to failure is the time until a failure occurs, so you can say its the time where the software is working as intendent.

The time to repair is the time it takes to fix a failure, so its the time when the software can't be used. +

The time between failure is the time between the different failures, so if there is a failure after 2 seconds, that failure is repaired in 1 second (time is now 3s) and the software fails again when time is 6seconds the time between failures was 6-2 = 4seconds. +

One quality factor that can be estimated is reliability which is the probability that the software executes with no failure during a certain time period. This can be estimated by calculating the mean-value of time to failure (=MTTF) and then: $R = \frac{MTTF}{(MTTF+1)}$ +3

/8

LINKÖPINGS UNIVERSITET

AID-nummer: *AID-number:* 1778	Datum: *Date:* 16-10-28	Blad nummer: *Sheet number:* 8
Kurskod: *Course code:* TDDC88	Provkod: *Exam code:* TEN1	

6. For project i) I will fix the parameters "feature" and "quality", I will do this because the system is safty-critical which means that it is important that it really lives up the quality constraints that are specified and also that all identified features needed are implemented. why? A system like that can run into truble if some quality constraints was needed to be removed in case of time running out. If there would be any trouble in the project it would be possible to postpone the release in order to get it work properly. I think that would be more appriciated than fail to meet other requirements. And if the time is critical it would be possible to add more resources if needed, although some consideration is needed if the resourses are added late in the project. 4/

For project ii) I would instead fix the parameters "time" and "resources". I would do this because I think this kind of product would be good to do within the planned time in order to reach the market before somcone else does, and in this case I think it would be more suitable to let the features & quality vary and that you start with the most prioritized features. In an app like that there would be many possible features (and levels of quality) and may not be such a harm if everything isn't implemented. The resources would be fixed, because I don't think you would like to spend too many resources on a project like this. If the project comes into buble, like that the time is running out, you can just skip implementing the less important features. 5/9

7. a) One activity I will use is "prototyping" and I will use so called "throw away prototyping" meaning that prototypes will be created quit fast, shown to the users which then can give feedback, based on this feedback some parts will be implemented, and new prototypes will be created and shown to the user and so on. The resauces needed is someone who does the prototypes and the material needed for that, as well as some users that gives feedback, so it involves perhaps a designer and the end-users. This activity can be used in the design phases of the projects. This will be good for the usability because it involves the user and let them give feedback which then will improve the usability because the needs of the user will be expressed and in the end the usability is something that the users determine.

8

b)
- Description: Number of good and bad feature according to the users.
- How to get data: Set up a test senario that is then done by some users. Afterwards I will ask them questions about the features and if they were good or bad.
- How to calculate: I will calculate an average of good features and one average of bad features. So I will obtain two values.
- Many good and few bad feature indicates high usability, because this means that the system fits well to the users needs, so it means that the relevance of the system is high which also gives a high usability.

4

7b) An other metrics can be numbers of calls for support
The data I will obtain by register all calls from the users for support for each day for some weeks or months.

The metric will then be calculated as the average number of calls per day.

A high value of this metrics will indicate a bad usability, though it indicates that the system can't be used in an efficient way and the fact that the system isn't very learnable because they need support even though they have had a one-week of education. 4

The third metric I will use is the avarage of the "customer happines" (kundnöjdhet). I will get data by sending out a form where the users are asked to rate how happy they are with the system in a scale from 1 to 5.

The metric will then be calculated by taking the average of the rates.

A high value of this metric will indicate a good usability because if the system is usable it is more likely that they are happy with the system and rate it high. 3

19

LINKÖPINGS UNIVERSITET

AID-nummer: *AID-number:* 1778	Datum: *Date:* 16-10-28	Blad nummer: *Sheet number:* 11
Kurskod: *Course code:* TDDC88	Provkod: *Exam code:* TEN1	

8.

- Insignificant: The coffee-machine in the personal-cafeteria is broken which makes the developers unhappy and hence may slow down the efficiency. Well, I don't think this will have much impact of the outcome and hopefully everyone doesn't drink coffee and can keep up the good work. If this occurs I just calls for the service and get the machine working again. +

- Tolerable: The person who should held the education gets sick at the day it should take place. This is tolerable because there are others who can give the lesson as well, although they aren't that skilled in "how to teach". If this occurs I will send someone else to do the education and give the customer the possibility to get one or two days extra education if they feel like they need to. ++

- Serious: The database-expert responsible for develop and designing the database quits in the middle of the project. This risk is serious because without this person it is very likely that the project will be delayed and also that the quality will decrease. If this happens I will beg the person to stay and offer an extra bonus on top of this months sulary. ++

- Catastrophic: The system crasches so badly that the customer can't use it for a quite long time. This is like a disaster because if the customer can't use the system

LINKÖPINGS UNIVERSITET

AID-nummer: *AID-number:* 1778	Datum: *Date:* 16-10-28	Blad nummer: *Sheet number:* 12
Kurskod: *Course code:* TDDC88	Provkod: *Exam code:* TEN1	

they will probably loose some of their customers and our company will not be very popular. If our customer In worst case they goes to someone else. tell this story to others our image can be detroyed as well. If this happens I will put in all my efforts and resources to make the system work again. ++

Sensible risk +1

/9

LINKÖPINGS UNIVERSITET

AID-nummer: *AID-number:* 1778	Datum: *Date:* 16-10-22	Blad nummer: *Sheet number:* 13
Kurskod: *Course code:* TDDC88	Provkod: *Exam code:* TEN1	

9. (I though I should need more space, please turn the paper to be able to see the diagram better)

•The user sends its username and password via the browser to the Facebook service. If the credentials are valid the users page, called "my page" is sent as a return-message and then shown to the user by the browser. The user than tries to post a picture. The file is passed by the browser to Facebook, if the user has access the picter editing mode is returned and showed to the user. Then a loop starts that continues as long as the user isn't satisfied with the picture. In the loop the user edits the file and after each edit the "new" picture is returned and shown. When the user is satisfied the loop ends. Then the description window is passed to the user and the user answers with the text. When the user press the return button this is "sent" to Facebook that then uploads the file. (This is the branch [access]) If the user don't have access to upload a file, the request is rejected. Then the user press logout and the user is logged out and the Facebook "start page" is browsed.