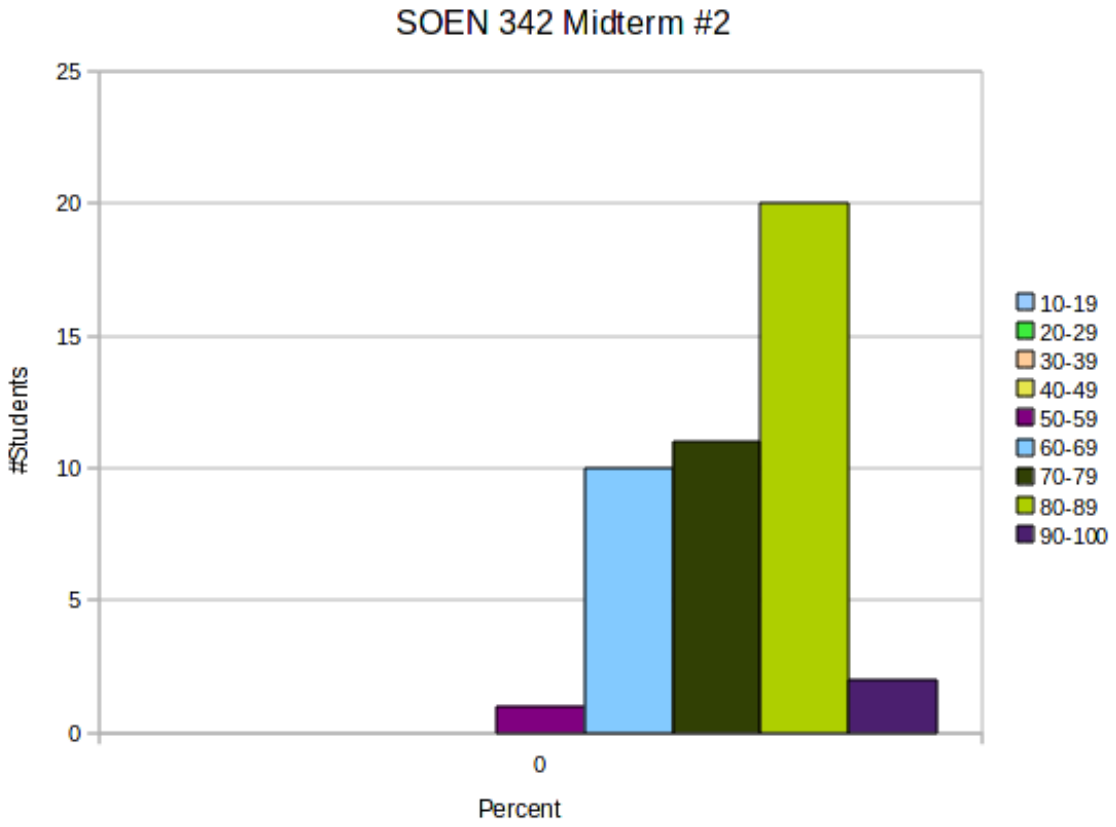
SOEN 342 Software Requirements

Fall 2009/2010

Midterm Exam #2

Name: _____ Total Points:

ID: _____ _____ / 25



	Q1	Q2	Q3	Q4	Total
Average	5.58	4.13	4.73	4.82	19.25
Percent	79.71	68.75	78.79	80.3	77.00

(7pts) **1.** Consider the following domain description for email clients:

7 pts

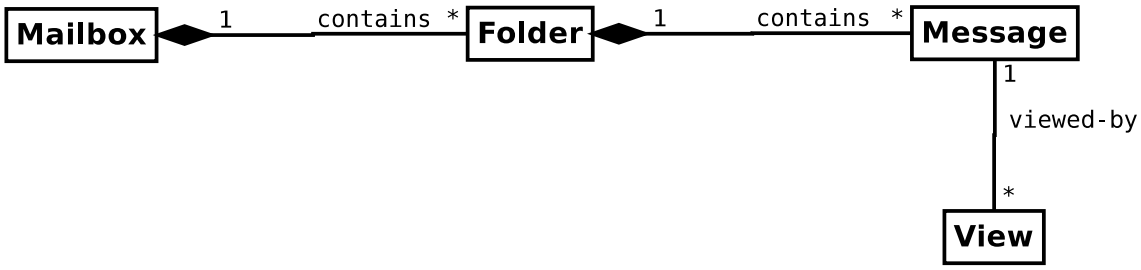
> The client has one mailbox which consists of a number of different folders. Each folder contains a number of messages. A *message* cannot exist in more than one folder and cannot exist outside a folder. A user can invoke a view on a message and in fact a user may have multiple views, each corresponding to a single message.

(a) (1 pt) Name an appropriate method for identifying *conceptual classes* in this domain description: **Noun Phrase Identification**

(b) (1 pt) Use the method to create a list of domain concepts based on the provided description:

- **Mailbox**
- **Folder**
- **View**
- **Message**

(c) (4 pts) Create a domain model for the email client as a UML class diagram. Make sure you show all appropriate details, including associations, multiplicities, and aggregations.



(d) (1 pt) Illustrate the difference between *aggregation* and *composition* using your domain model. Give a brief explanation for each:

Aggregation: **whole-part associations, e.g., user–mailbox**

Composition: **aggregation without shared instances, e.g., message–folder**

Solution: See above. For (a), *"Use a conceptual class category list"* was also accepted.

Marking: For (b), you needed at least the concepts Folder–View–Message to get full marks.

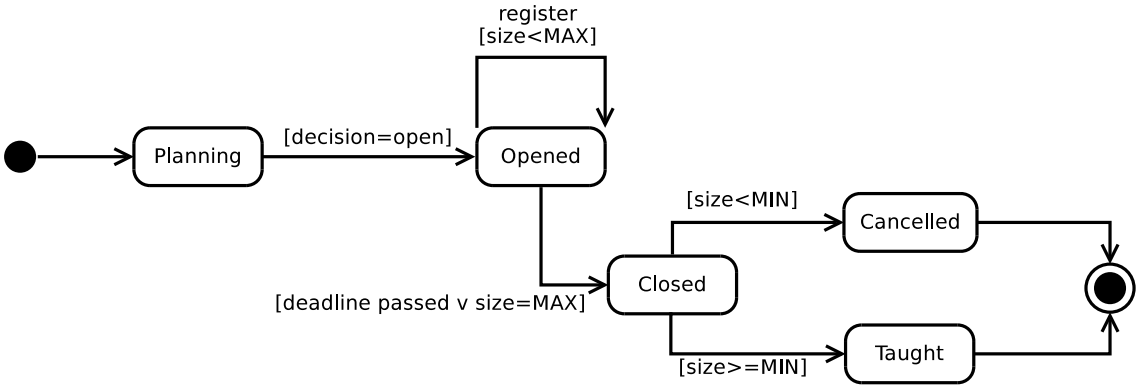
For (c), marks were distributed as follows: 1pt. for (at least) the conceptual classes Folder–View–Message; 1pt. for correct associations (no marks off if you forgot to label the associations); 1pt. for showing correct aggregations/compositions; and 1pt. for correct multiplicities. For minor mistakes/omissions, $\frac{1}{2}$ pt. was taken off.

(6pts) **2.** Consider the following specification for a course planning system:

6 pts

> The course section life cycle starts from its planning. Once the decision for opening the registration for the course is received, the course is opened. While the course is opened, the requests for registering can be accepted. The course will not be actually taught until the class size reaches a certain minimum. The requests to register are accepted until the course reaches the predefined maximum number of students, or the registration deadline has passed; in both cases, the course section becomes closed. If the class size is below the minimum, the class is cancelled. Closing the section when there are not enough students will have the same effect as cancelling it.

(a) (4 pts) Draw an UML state diagram to specify Course Section behaviour. Use hierarchical states where appropriate:



(b) (2 pts) *True* or *False*? ***Note:*** Each correct answer gives you $\frac{1}{2}$ points, each wrong answer $-\frac{1}{2}$, "don't know" 0 (however, you will never get a negative score for this question).

State machines can be used to describe legal system event within a use case.

☑ True ☐ False ☐ Don't know

Activity diagrams are useful to model use cases with many alternative flows or extensions.

☑ True ☐ False ☐ Don't know

State machines are an alternative modeling technique to domain models in RE.

☐ True ☑ False ☐ Don't know

State diagrams can show how a *single* object behaves across *different* use cases.

☑ True ☐ False ☐ Don't know

Marking: For (a), two points for correctly identified states and two points for correct transitions. No points off for variations, as long as the main requirements were correctly rerepsented (like not being able to register when a course is full). Points were taken off for missing or incorrect guards on transitions.

(6pts) **3.** Consider the following Z schema specification for a birthday book application:

6 pts

$[NAME,\ DATE]$

BirthdayBook

$known : \mathbb{P} NAME$

$birthday : NAME \pfun DATE$

$known = \mathrm{dom}\ birthday$

(a) (3 pts) Write a non-robust Z schema for the *UpdateBirthday* operation, which changes the date of an *existing* entry (i.e., if a name is not in the system, it will not be added by the UpdateBirthday operation).

UpdateBirthday

$\Delta BirthdayBook$

$name? : NAME$

$date? : DATE$

$name? \in known$

$birthday' = (birthday - \{name? \mapsto birthday(name?)\}) \cup \{name? \mapsto date?\}$

(b) (1 pt) Now make the operation robust by using the following two schemas for error handling:

$REPORT ::= ok \mid already\_known \mid not\_known$

Success

$result! : REPORT$

$result! = ok$

NotKnown

$\Xi BirthdayBook$

$name? : NAME$

$result! : REPORT$

$name? \notin known$

$result! = not\_known$

Define a robust version of *UpdateBirthday* that returns *ok* in case of success and *not_known* in case of an error:

$RUpdateBirthday =$ $(UpdateBirthday \land Success) \lor NotKnown$

(c) (2 pts) Now show the combined schemas for the *RUpdateBirthday* operation:

RUpdateBirthday

$\Delta BirthdayBook$

$name? : NAME$

$date? : DATEresult! : REPORT$

$(name? \in known \land$

$birthday' = (birthday - \{name? \mapsto birthday(name?)\}) \cup \{name? \mapsto date?\} \land$

$result! = ok) \lor$

$(name? \notin known \land birthday' = birthday \land result! = not\_known)$

Marking: For (a), 1pts. each for correct state variables, pre- and postcondition ($\frac{1}{2}$ marks off for minor mistakes). A common mistake was to forget to remove the old mapping $\{name? \mapsto birthday(name?)\}$, but no points were taken off for this error.

For (c), one pts. each for correctly combined state variables and assertions. A common omission was to not specify $birthday'$ in case $name? \notin known$, but not points were taken off for this error.

(6pts) **4.** Consider the *"Display a list of sales regions"* use case of a simple system for managing sales regions:

6 pts

1. This use case starts when the user selects to display the list of sales regions.
2. The system displays the list of sales regions (region code and region name) in alphabetical order of region names.
3. If there is no region in the list, the system displays "No region available" in the region name.

Apply the COSMIC-FFP measurement method to compute the functional size.

(a) (1 pt) Identify the triggering event:
the user selects to display the list of sales regions

(b) (3 pts) Use COSMIC-FFP to model the functional process. Insert the values into the table below:

ID Process	Process Description	Data Movement	Data Group	Data Movement Type	CFP
1.1	Display Sales Regions	Request	Request Event	E	1
		Extract the region codes and names	SalesRegions	R	1
		(LOGIC) order the list by name			
		Display the list of sales regions	SalesRegions	X	1
		displays "No region available"	Error message	X	1

(c) (1 pt) Calculate the total functional size of the "Display a list of sales regions" use case: **4**

(d) (1 pt) How can you use (COSMIC) Function Points for cost estimation?
Using additional CFP–effort data, e.g., derived from previous projects.

Marking: For (b), $\frac{1}{2}$–1 marks were taken off for various mistakes, such as not correctly identifying the data groups or movement types.

— End of Exam Solutions —