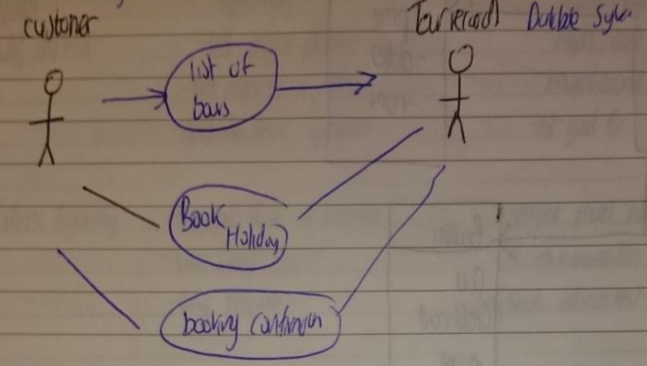
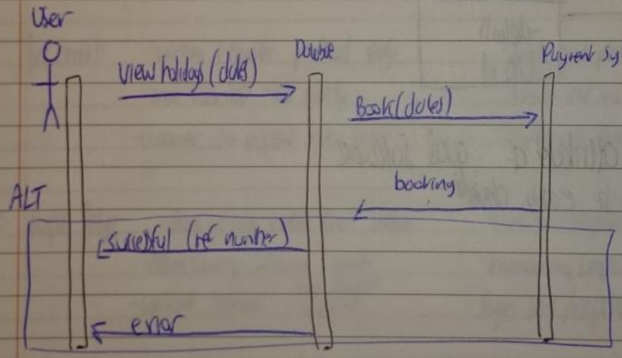
SOFTWARE ENGINEERING

12/04/15. EXAM PAPER 2014

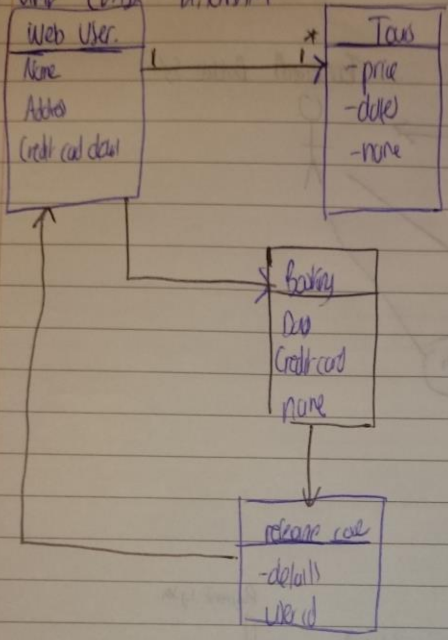
1 a. UML Case Diagram



UML Sequence Diagram

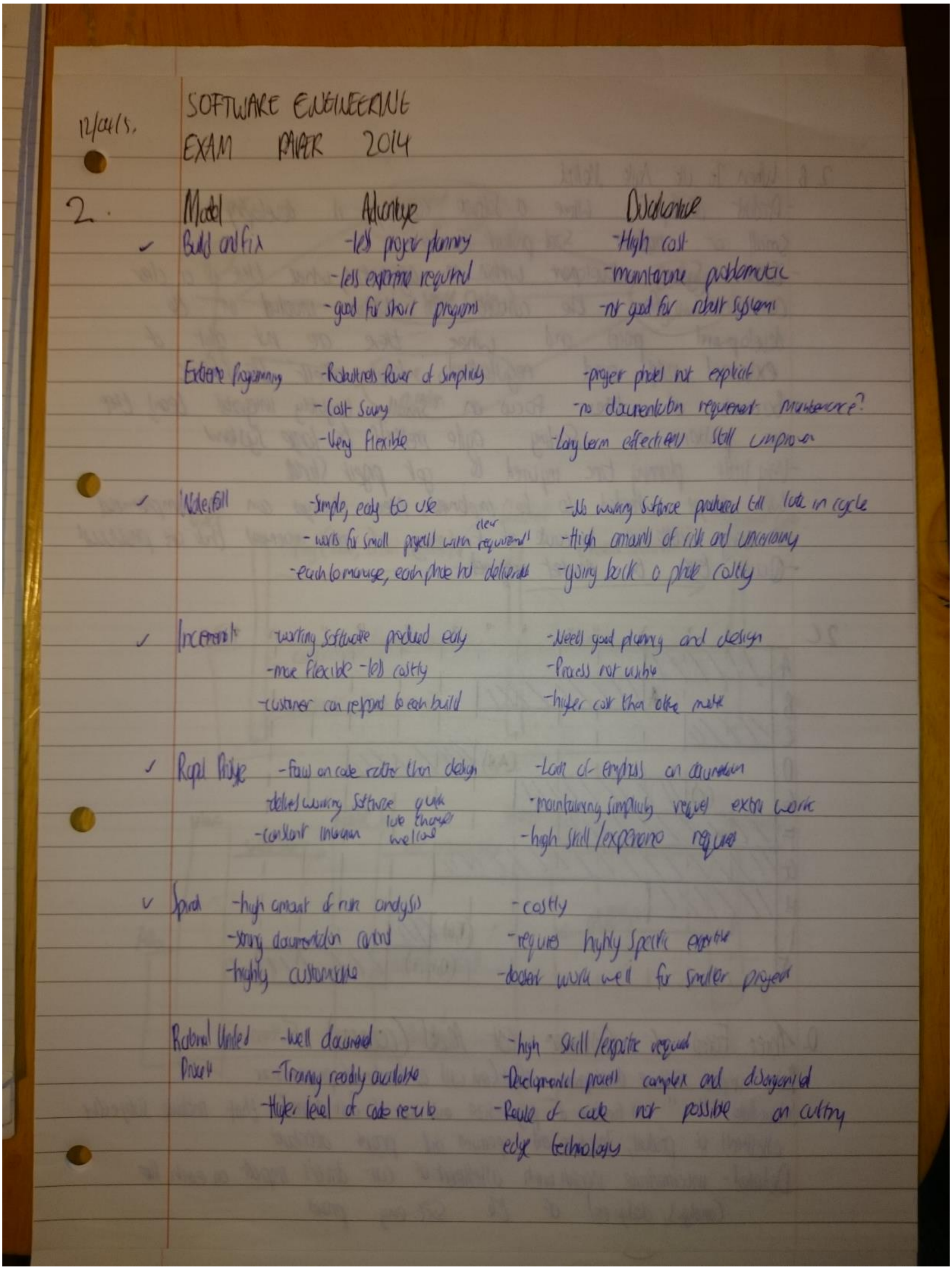


C UML CLASS DIAGRAM



D. Three essential attributes of good Software:
- Maintainability can be easily changed
- Dependability
- Efficiency
- Acceptability

E For Fundamental Software engineering activities
- Software specification
- Software development
- Software validation
- Software evolution



12/04/15.

SOFTWARE ENGINEERING
EXAM PAPER 2014

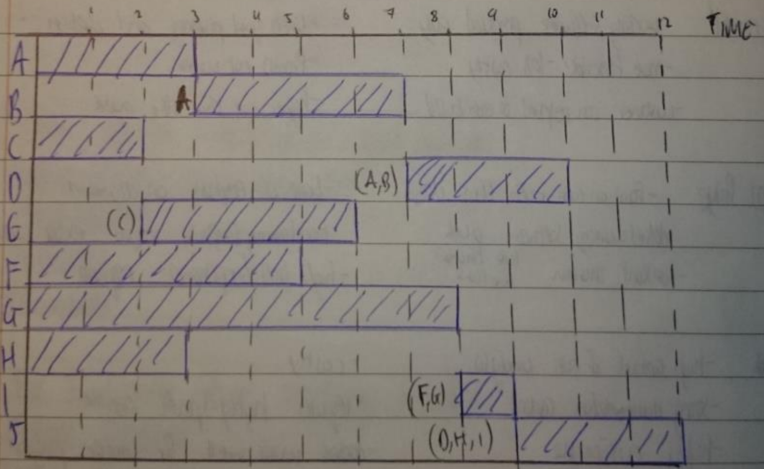
2.

	Model	Advantage	Disadvantage
✓	Build and fix	-less project planning -less expertise required -good for short programs	-High cost -maintenance problematic -not good for robust systems
	Extreme Programming	-Robustness-Power of simplicity -Cost saving -Very flexible	-project phases not explicit -no documentation requirement maintenance? -Long term effectiveness still unproven
✓	Waterfall	-Simple, easy to use -works for small projects with clear requirements -easy to manage, each phase has deliverables	-No working software produced till late in cycle -High amounts of risk and uncertainty -going back a phase costly
✓	Incremental	working software produced early -more flexible -less costly -customer can respond to each build	-Needs good planning and design -Process not visible -higher cost than other models
✓	Rapid Prototype	-focus on code rather than design delivers working software quick -constant involvement late changes welcome	-Lack of emphasis on documentation -maintaining simplicity requires extra work -high skill/experience required
✓	Spiral	-high amount of risk analysis -strong documentation control -highly customisable	-costly -requires highly specific expertise -doesn't work well for smaller projects
	Rational Unified Process	-well documented -Training readily available -Higher level of code reuse	-high skill/expertise required -Development process complex and disorganised -Reuse of code not possible on cutting edge technology

2 B When To use Agile Method.

- Product development where a Software company is developing a small or medium sized product for sale
- Custom System development within an organisation, where there is a clear commitment from the customer to become involved in the developmental process and where there are not alot of external rules and regulations that affect the Software
- Because of their focus on small, tightly integrated teams, there are problems in scaling agile methods to large systems
- Very little planning time required to get project started
- When changes needed to be implemented. New changes can be implemented at very little cost, because of frequency of new increments that are produced
- Quick time to market required

2C.

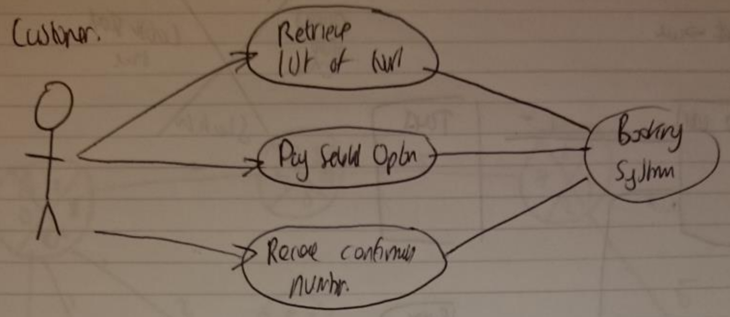


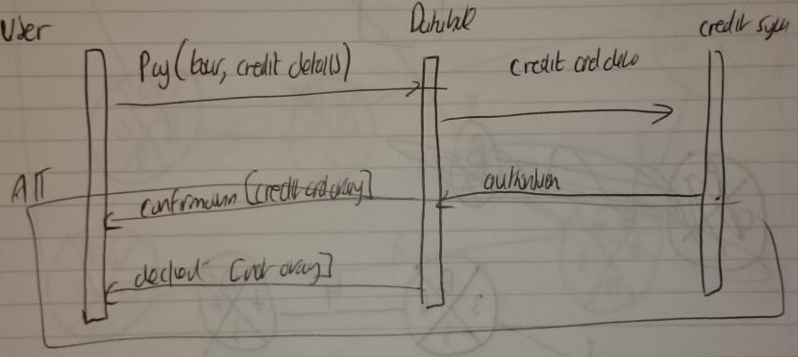
D. Three Forms of Constructive Cost Model (COCOMO)

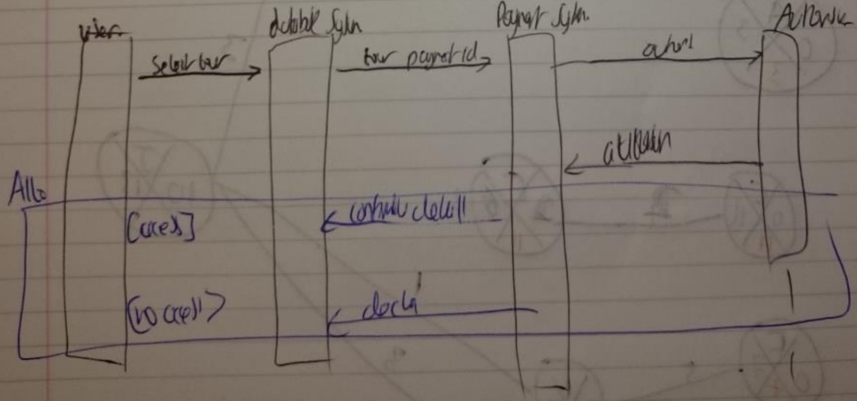
Basic - computes software development effort (and cost) as function of program size

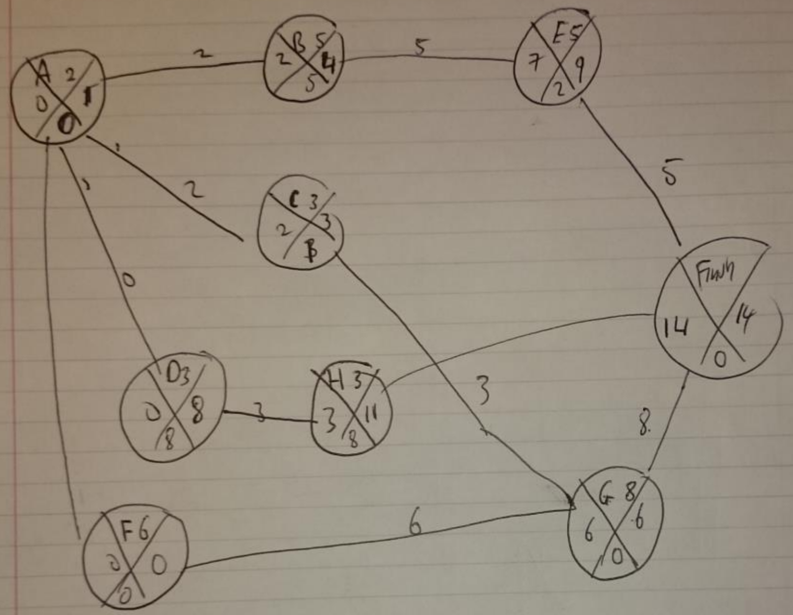
Intermediate - " " as function of program size and set of "cost drivers" that include subjective assessments of product, hardware and personell and project attributes

Detailed - intermediate version with assessment of cost driver's impact on each step (analysis, design etc) of the Soft. eng. process


Customer.
Retrieve list of tours
Pay Selected Option
Receive confirmation number.
Booking System


User
Database
credit system
Pay(tour, credit details)
credit card data
Alt
confirmation [credit card okay]
authorisation
declined [card okay]


User
database System
Payment System.
Authoriser
Select tour
tour payment id
auth
authorisation
Alt
[access]
confirm details
[no access>
decline


A 2
0
1
0
2
B 5
2
4
5
5
E 5
7
9
2
2
C 3
2
3
3
0
D 3
0
8
8
3
H 3
3
11
8
3
5
Finish
14
14
0
8.
F 6
0
0
0
6
G 8
6
6
0