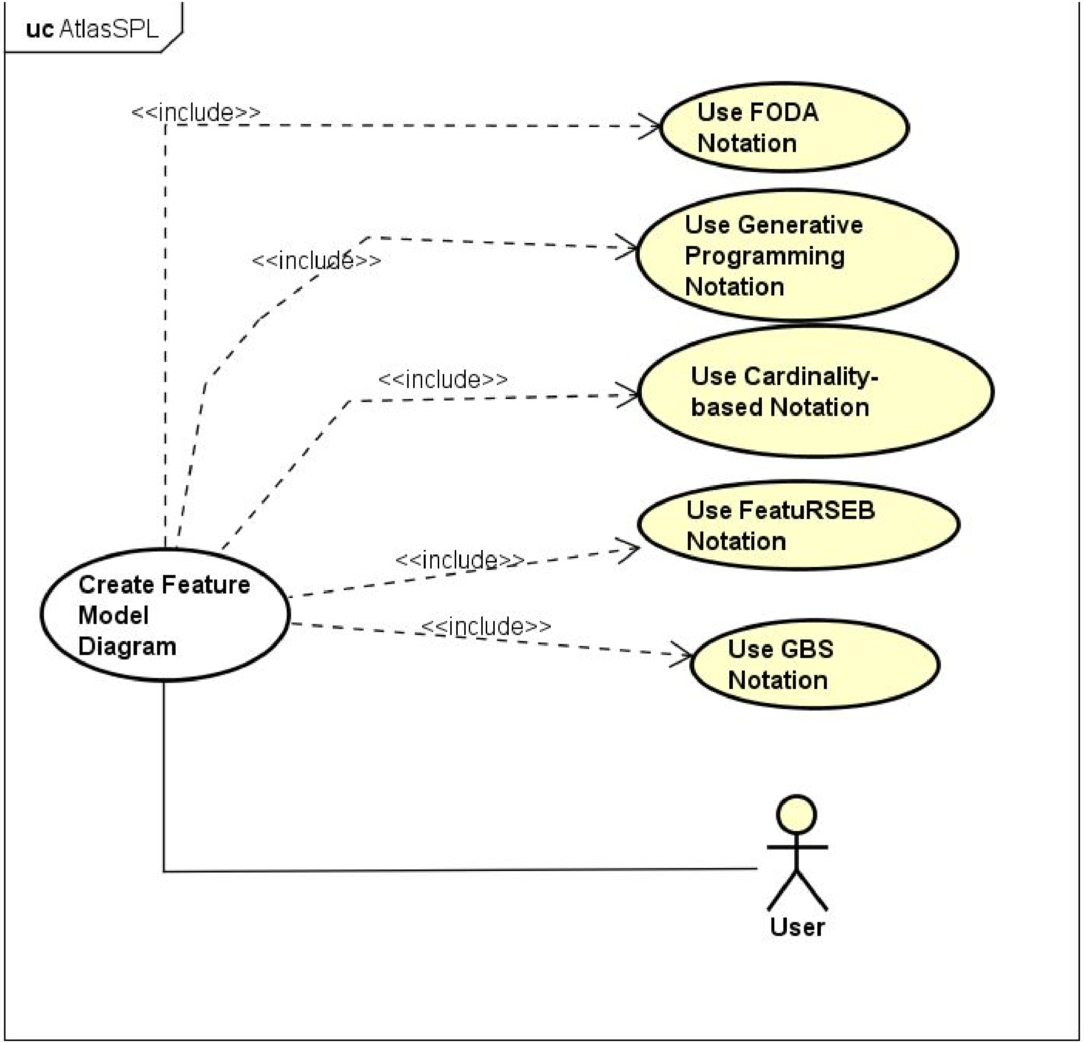
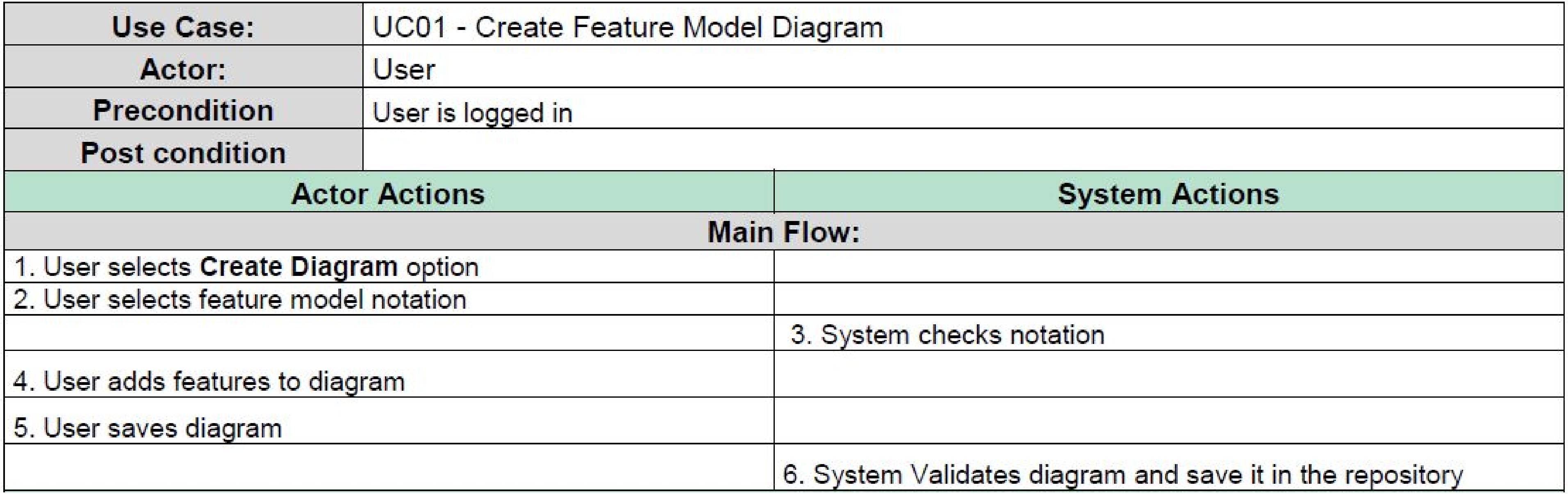
Requirement Analysis

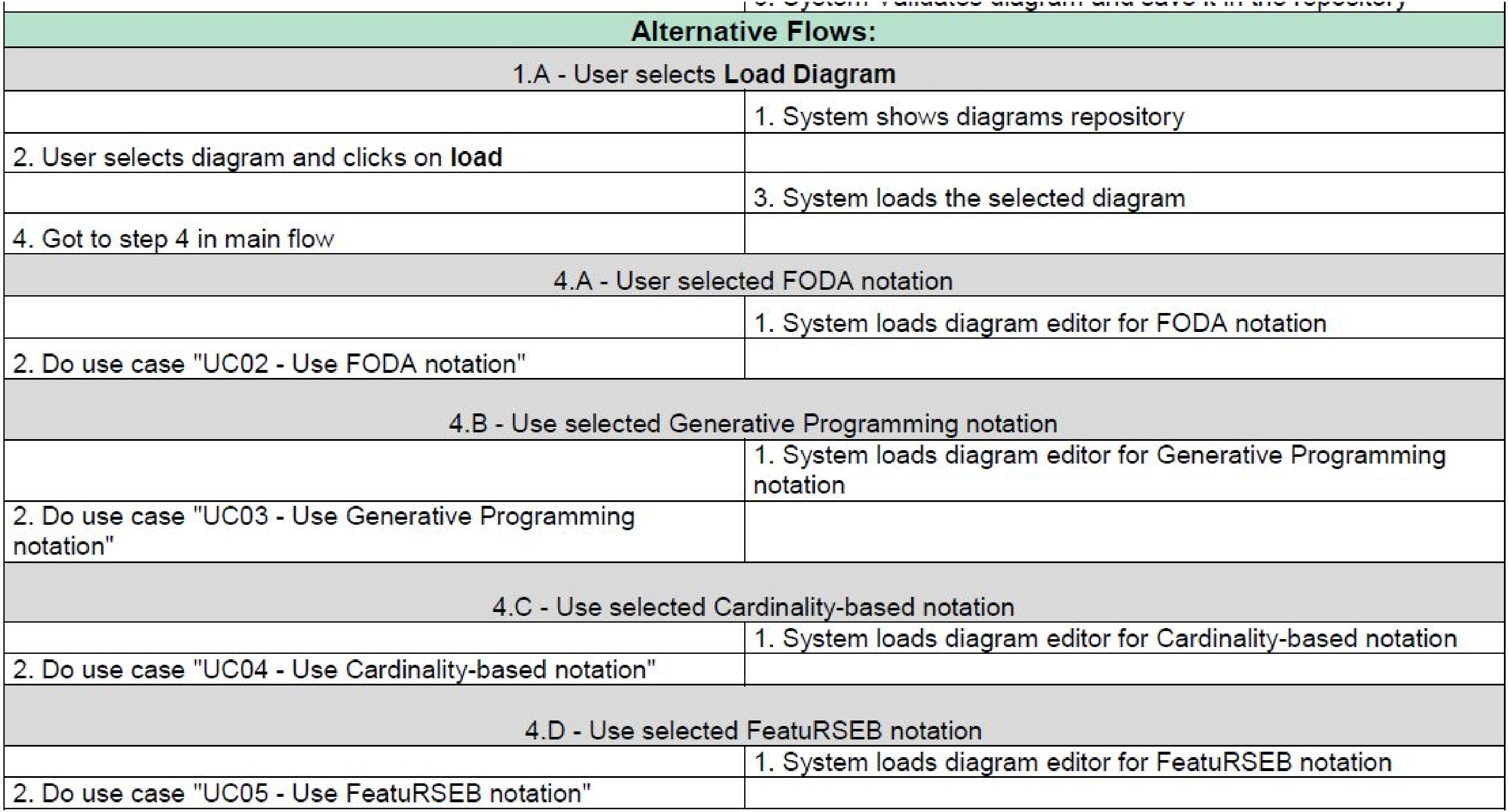
Luciano Marchezan

Use case diagram

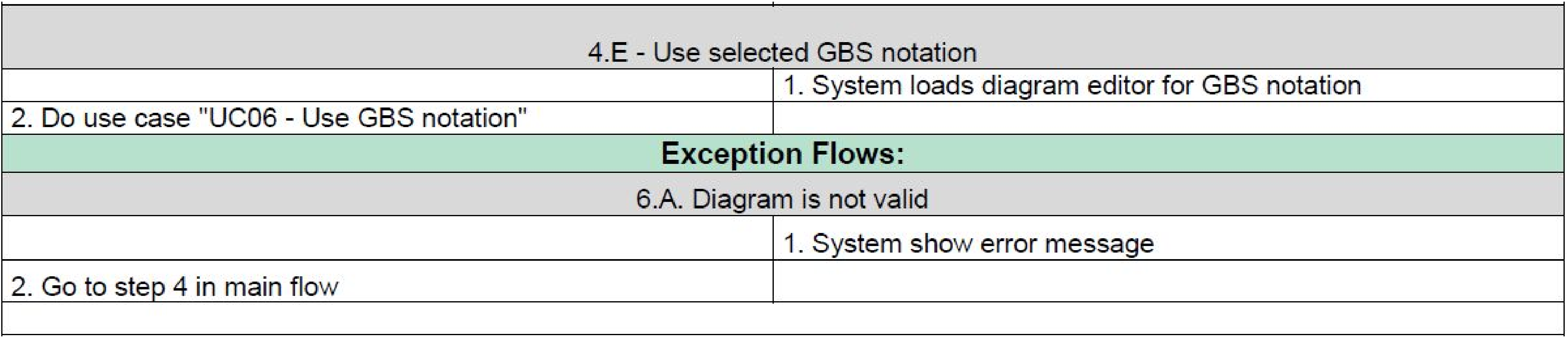




Use Case:	UC01 - Create Feature Model Diagram
Actor:	User
Precondition	User is logged in
Post condition	
Actor Actions	**System Actions**
Main Flow:	
1. User selects **Create Diagram** option	
2. User selects feature model notation	
	3. System checks notation
4. User adds features to diagram	
5. User saves diagram	
	6. System Validates diagram and save it in the repository

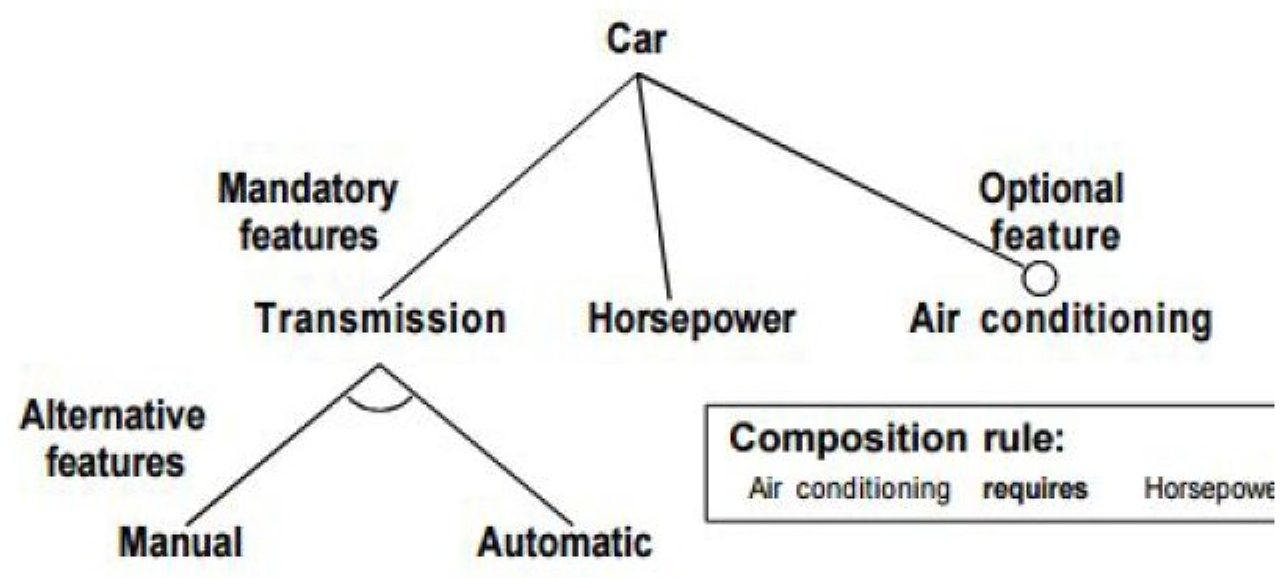


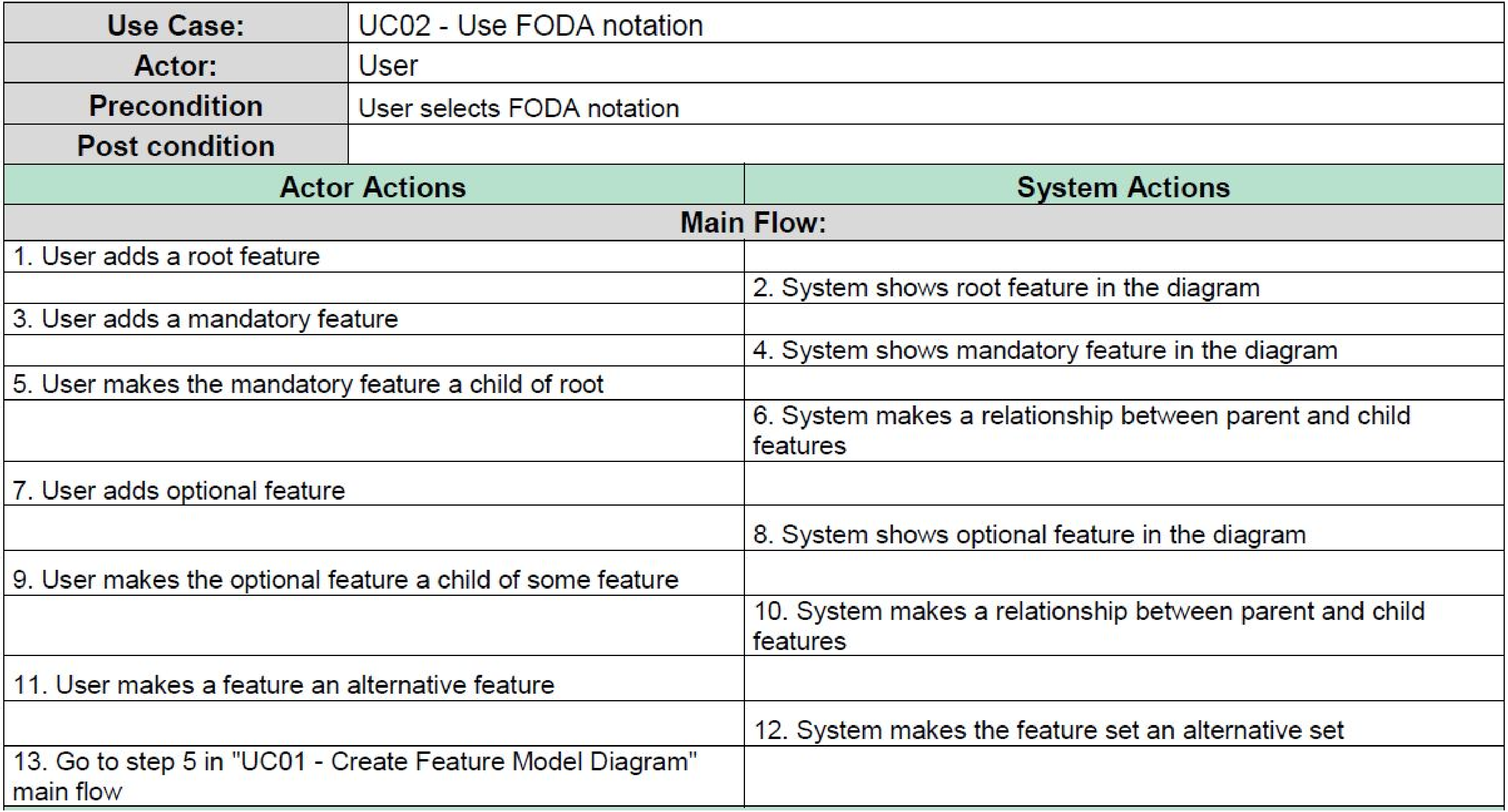
	6. System validates diagram and save it in the repository
Alternative Flows:	
1.A - User selects **Load Diagram**	
	1. System shows diagrams repository
2. User selects diagram and clicks on **load**	
	3. System loads the selected diagram
4. Got to step 4 in main flow	
4.A - User selected FODA notation	
	1. System loads diagram editor for FODA notation
2. Do use case "UC02 - Use FODA notation"	
4.B - Use selected Generative Programming notation	
	1. System loads diagram editor for Generative Programming notation
2. Do use case "UC03 - Use Generative Programming notation"	
4.C - Use selected Cardinality-based notation	
	1. System loads diagram editor for Cardinality-based notation
2. Do use case "UC04 - Use Cardinality-based notation"	
4.D - Use selected FeatuRSEB notation	
	1. System loads diagram editor for FeatuRSEB notation
2. Do use case "UC05 - Use FeatuRSEB notation"	



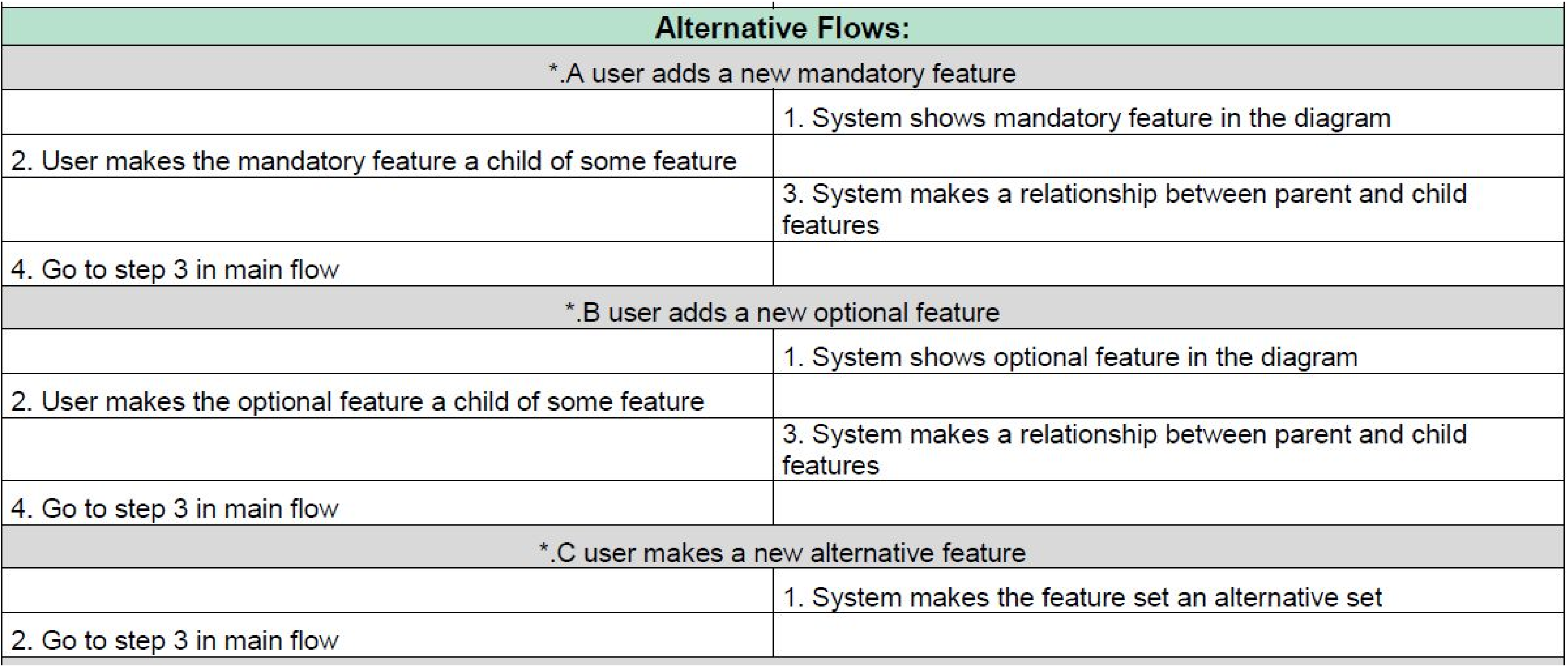
4.E - Use selected GBS notation	
	1. System loads diagram editor for GBS notation
2. Do use case "UC06 - Use GBS notation"	
Exception Flows:	
6.A. Diagram is not valid	
	1. System show error message
2. Go to step 4 in main flow	

FODA

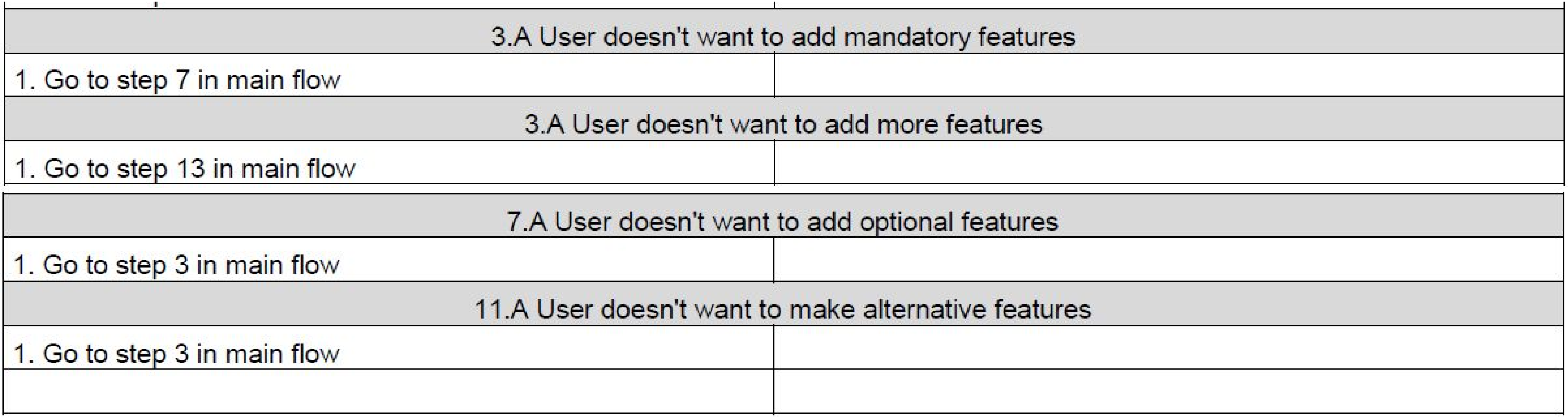

Car
Mandatory features
Optional feature
Transmission
Horsepower
Air conditioning
Alternative features
Manual
Automatic
Composition rule:
Air conditioning requires Horsepowe



Use Case:	UC02 - Use FODA notation
Actor:	User
Precondition	User selects FODA notation
Post condition	
Actor Actions	**System Actions**
Main Flow:	
1. User adds a root feature	
	2. System shows root feature in the diagram
3. User adds a mandatory feature	
	4. System shows mandatory feature in the diagram
5. User makes the mandatory feature a child of root	
	6. System makes a relationship between parent and child features
7. User adds optional feature	
	8. System shows optional feature in the diagram
9. User makes the optional feature a child of some feature	
	10. System makes a relationship between parent and child features
11. User makes a feature an alternative feature	
	12. System makes the feature set an alternative set
13. Go to step 5 in "UC01 - Create Feature Model Diagram" main flow	

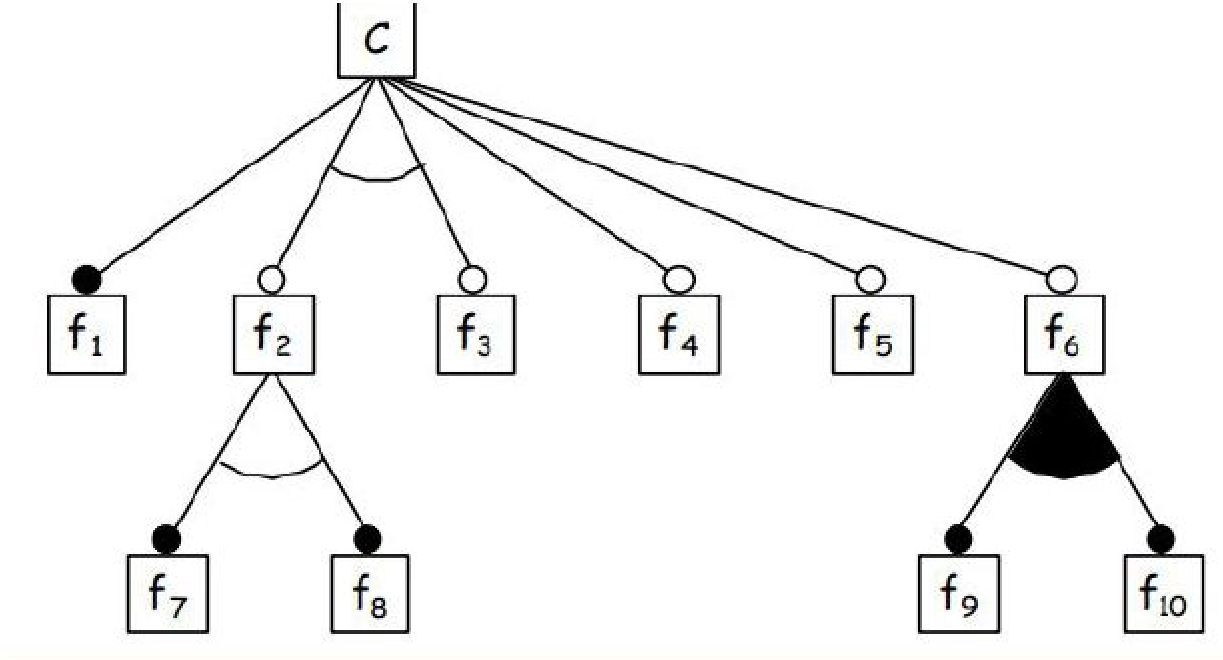


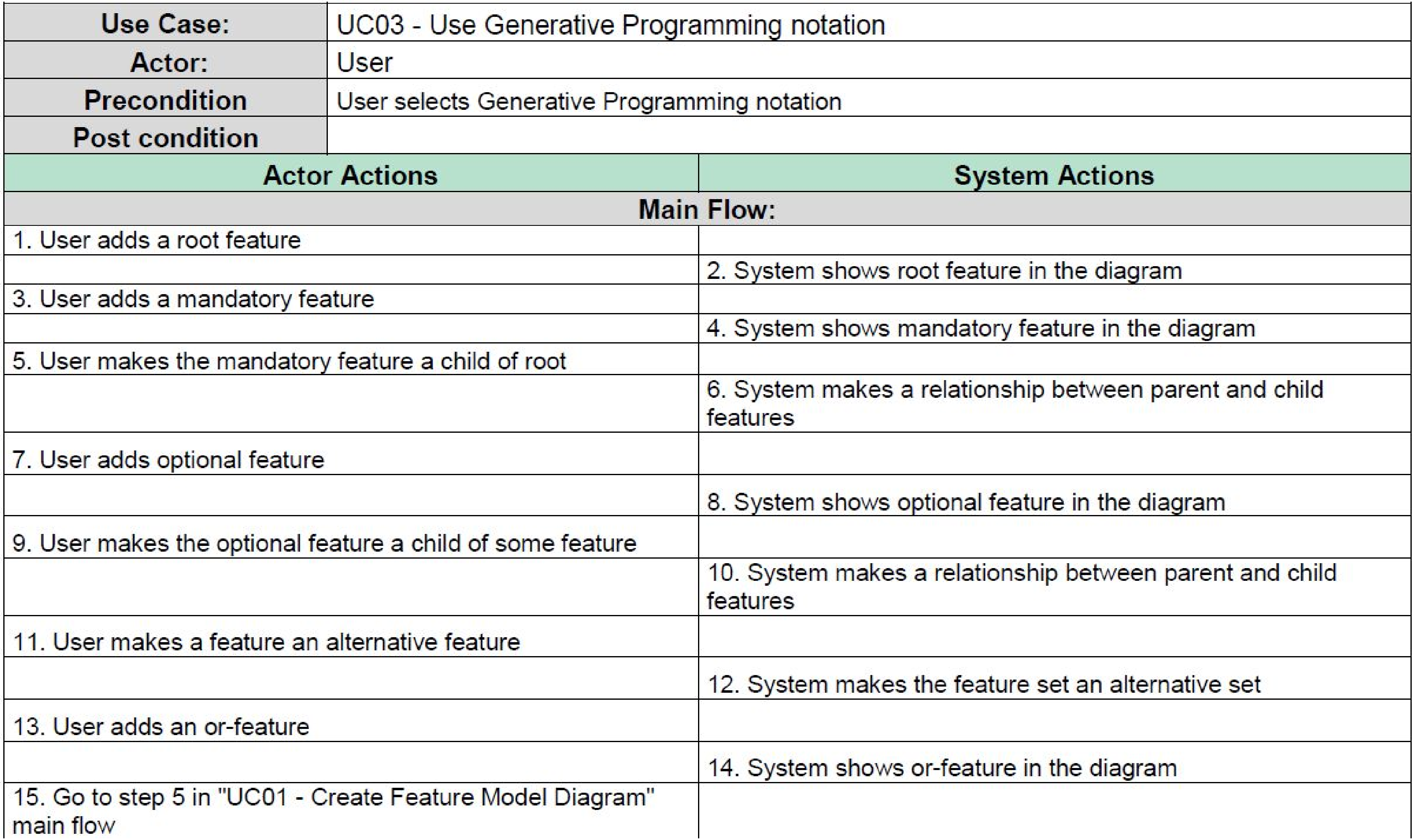
Alternative Flows:	
*.A user adds a new mandatory feature	
	1. System shows mandatory feature in the diagram
2. User makes the mandatory feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	
*.B user adds a new optional feature	
	1. System shows optional feature in the diagram
2. User makes the optional feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	
*.C user makes a new alternative feature	
	1. System makes the feature set an alternative set
2. Go to step 3 in main flow	



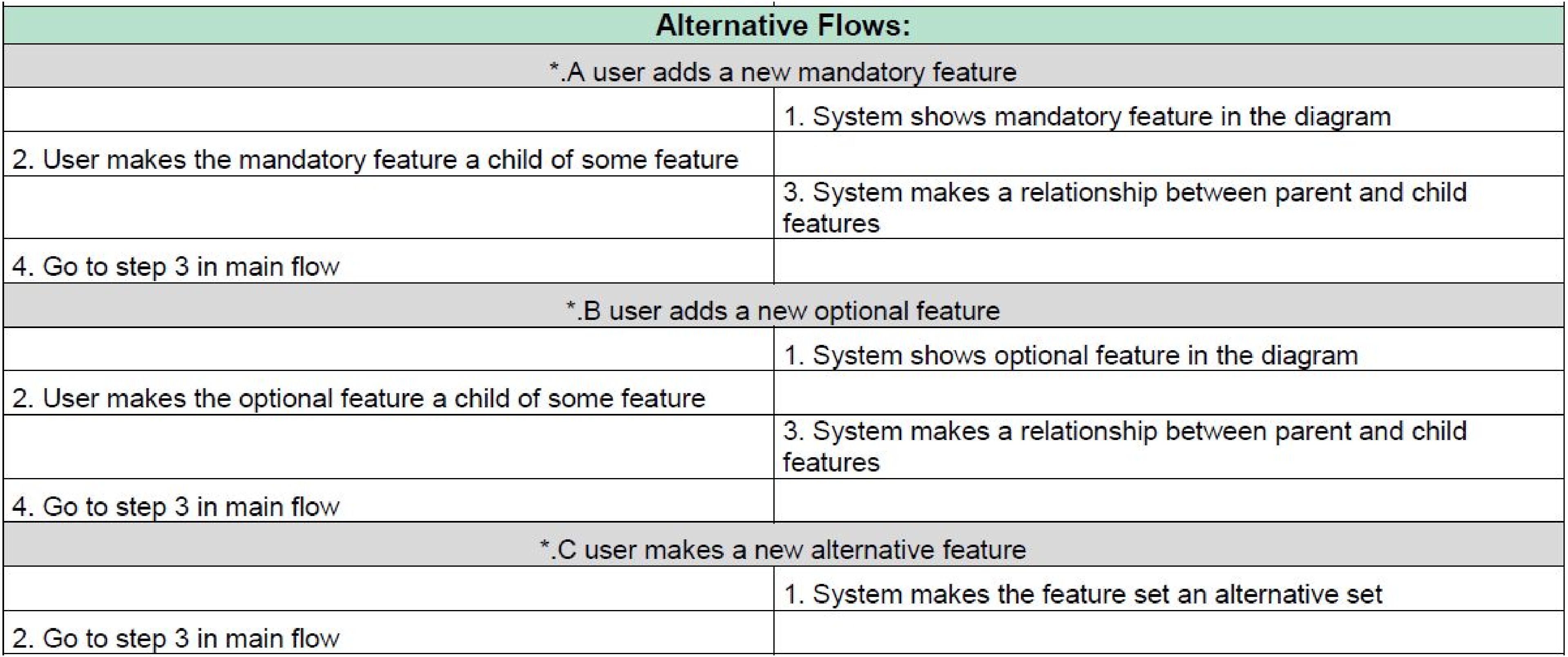
3.A User doesn't want to add mandatory features	
1. Go to step 7 in main flow	
3.A User doesn't want to add more features	
1. Go to step 13 in main flow	
7.A User doesn't want to add optional features	
1. Go to step 3 in main flow	
11.A User doesn't want to make alternative features	
1. Go to step 3 in main flow	

Generative Programming [Czarnecki 2000]

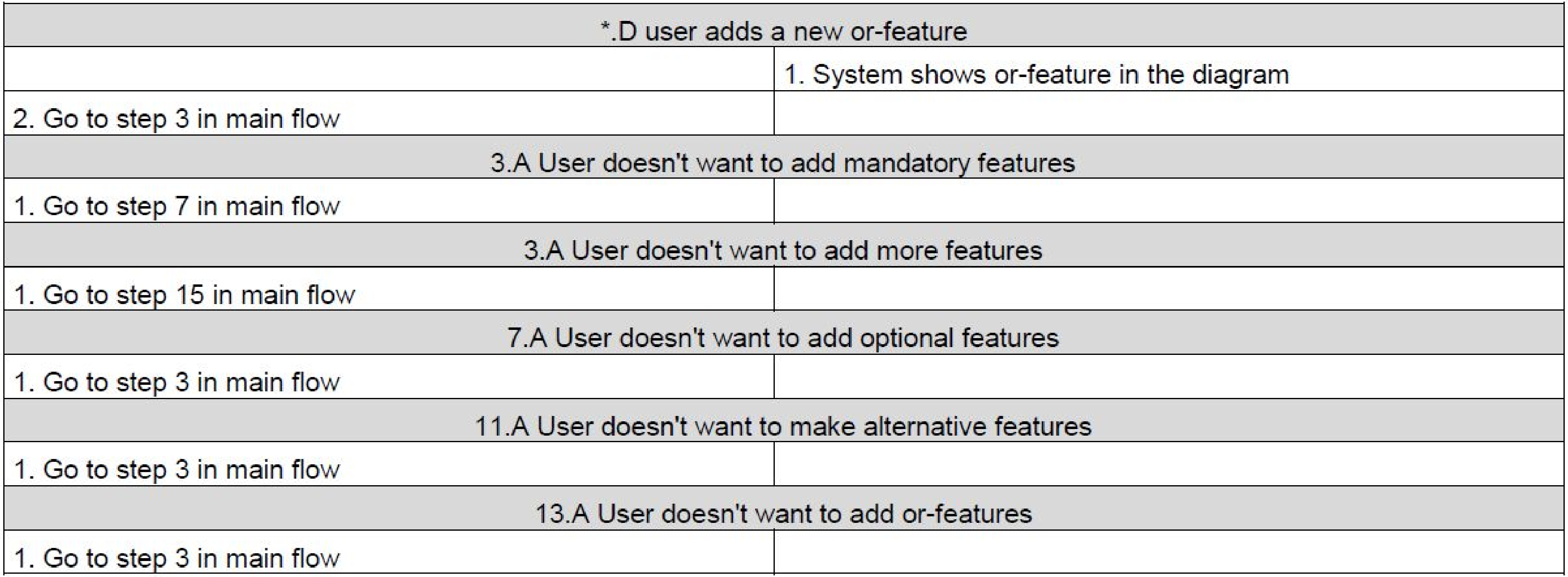




Use Case:	UC03 - Use Generative Programming notation
Actor:	User
Precondition	User selects Generative Programming notation
Post condition	
Actor Actions	**System Actions**
Main Flow:	
1. User adds a root feature	
	2. System shows root feature in the diagram
3. User adds a mandatory feature	
	4. System shows mandatory feature in the diagram
5. User makes the mandatory feature a child of root	
	6. System makes a relationship between parent and child features
7. User adds optional feature	
	8. System shows optional feature in the diagram
9. User makes the optional feature a child of some feature	
	10. System makes a relationship between parent and child features
11. User makes a feature an alternative feature	
	12. System makes the feature set an alternative set
13. User adds an or-feature	
	14. System shows or-feature in the diagram
15. Go to step 5 in "UC01 - Create Feature Model Diagram" main flow	

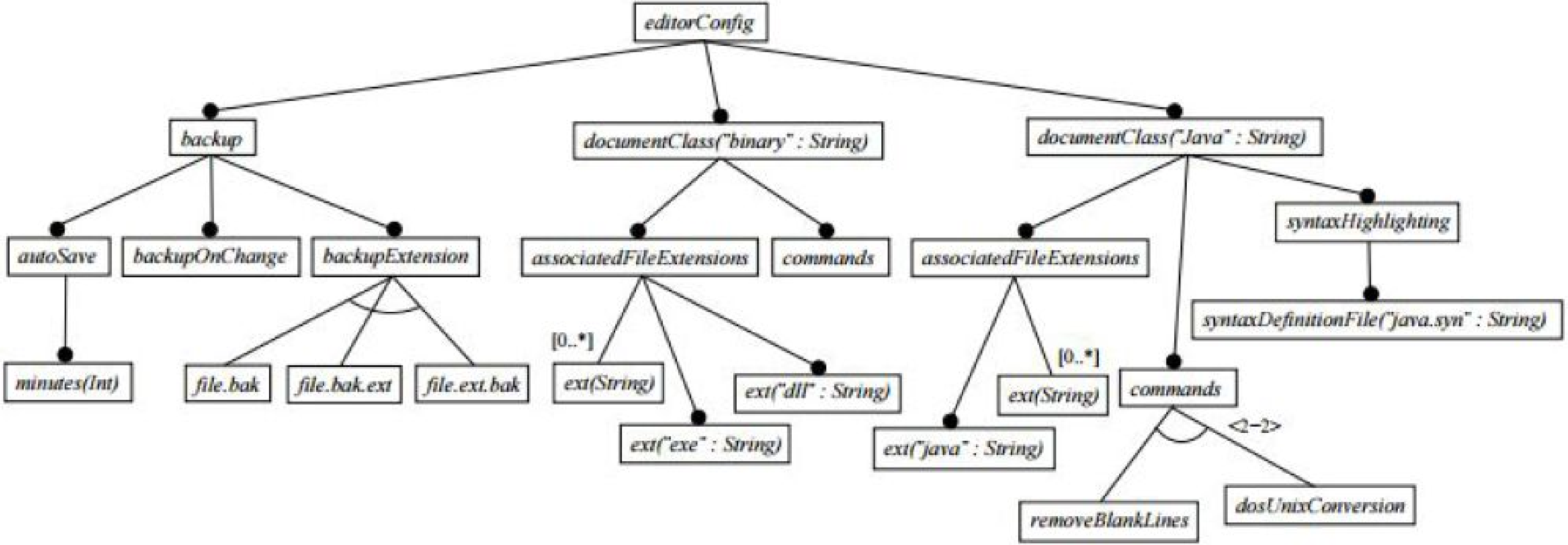


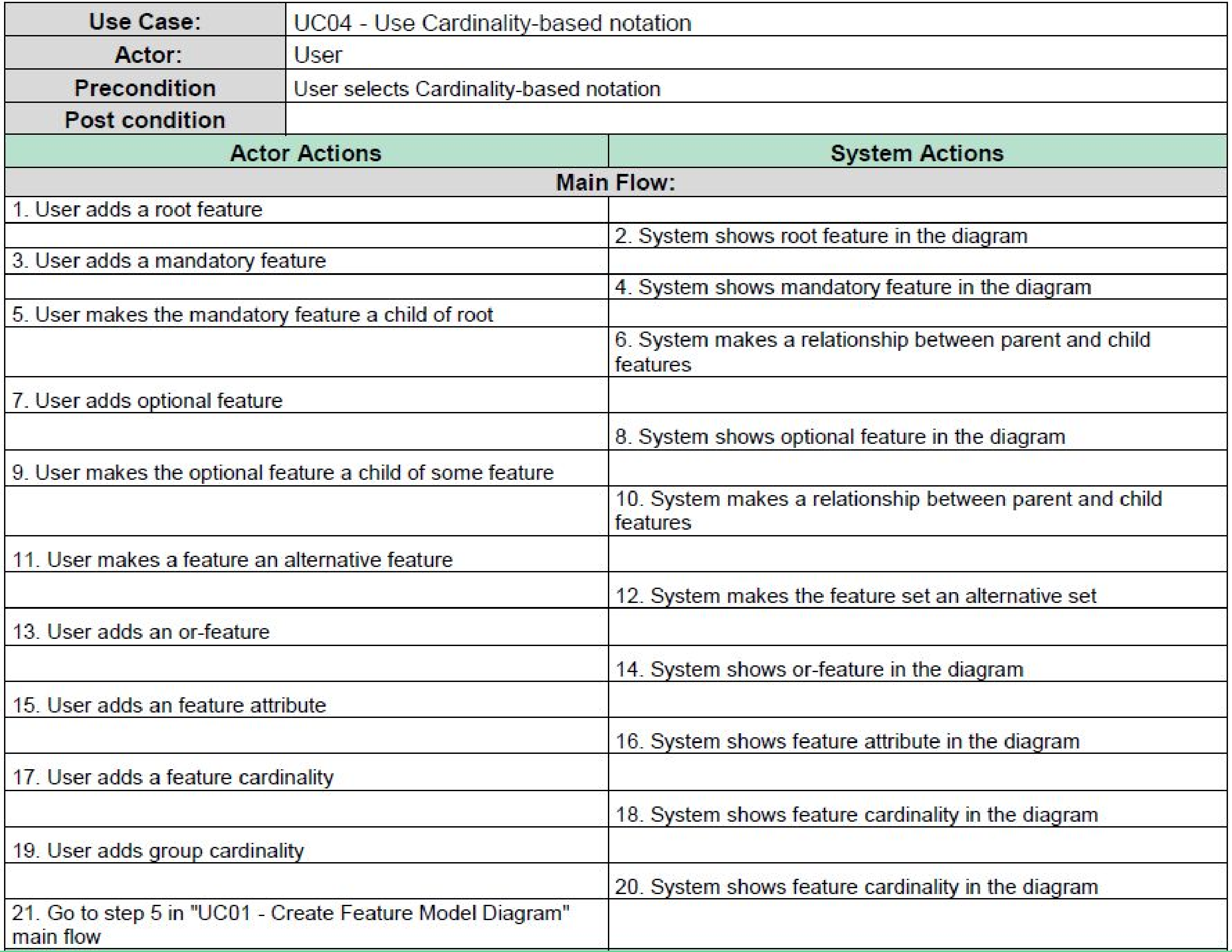
Alternative Flows:	
*.A user adds a new mandatory feature	
	1. System shows mandatory feature in the diagram
2. User makes the mandatory feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	
*.B user adds a new optional feature	
	1. System shows optional feature in the diagram
2. User makes the optional feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	
*.C user makes a new alternative feature	
	1. System makes the feature set an alternative set
2. Go to step 3 in main flow	



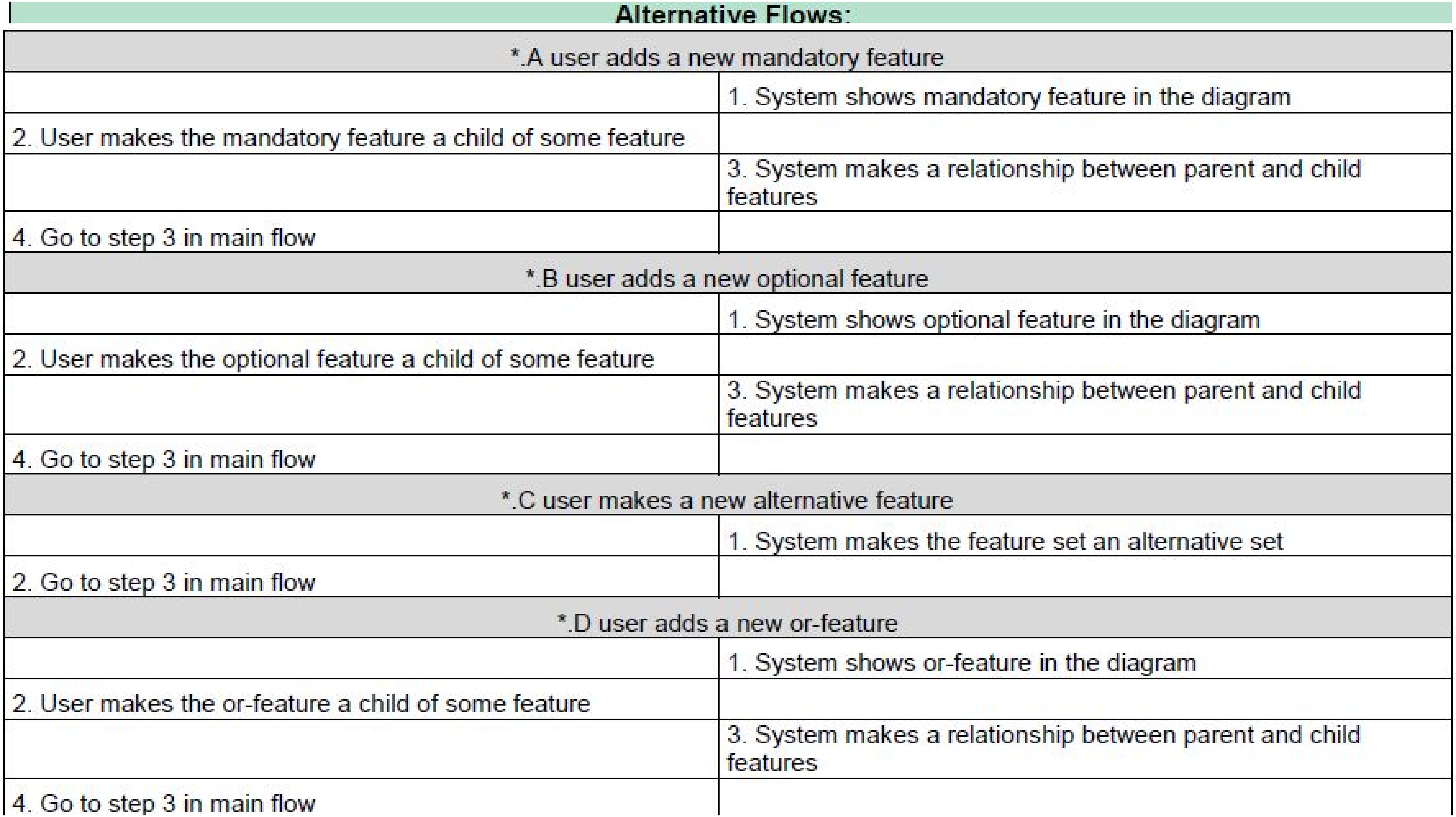
*.D user adds a new or-feature	
	1. System shows or-feature in the diagram
2. Go to step 3 in main flow	
3.A User doesn't want to add mandatory features	
1. Go to step 7 in main flow	
3.A User doesn't want to add more features	
1. Go to step 15 in main flow	
7.A User doesn't want to add optional features	
1. Go to step 3 in main flow	
11.A User doesn't want to make alternative features	
1. Go to step 3 in main flow	
13.A User doesn't want to add or-features	
1. Go to step 3 in main flow	

Cardinality-based [Czarnecki 2005]



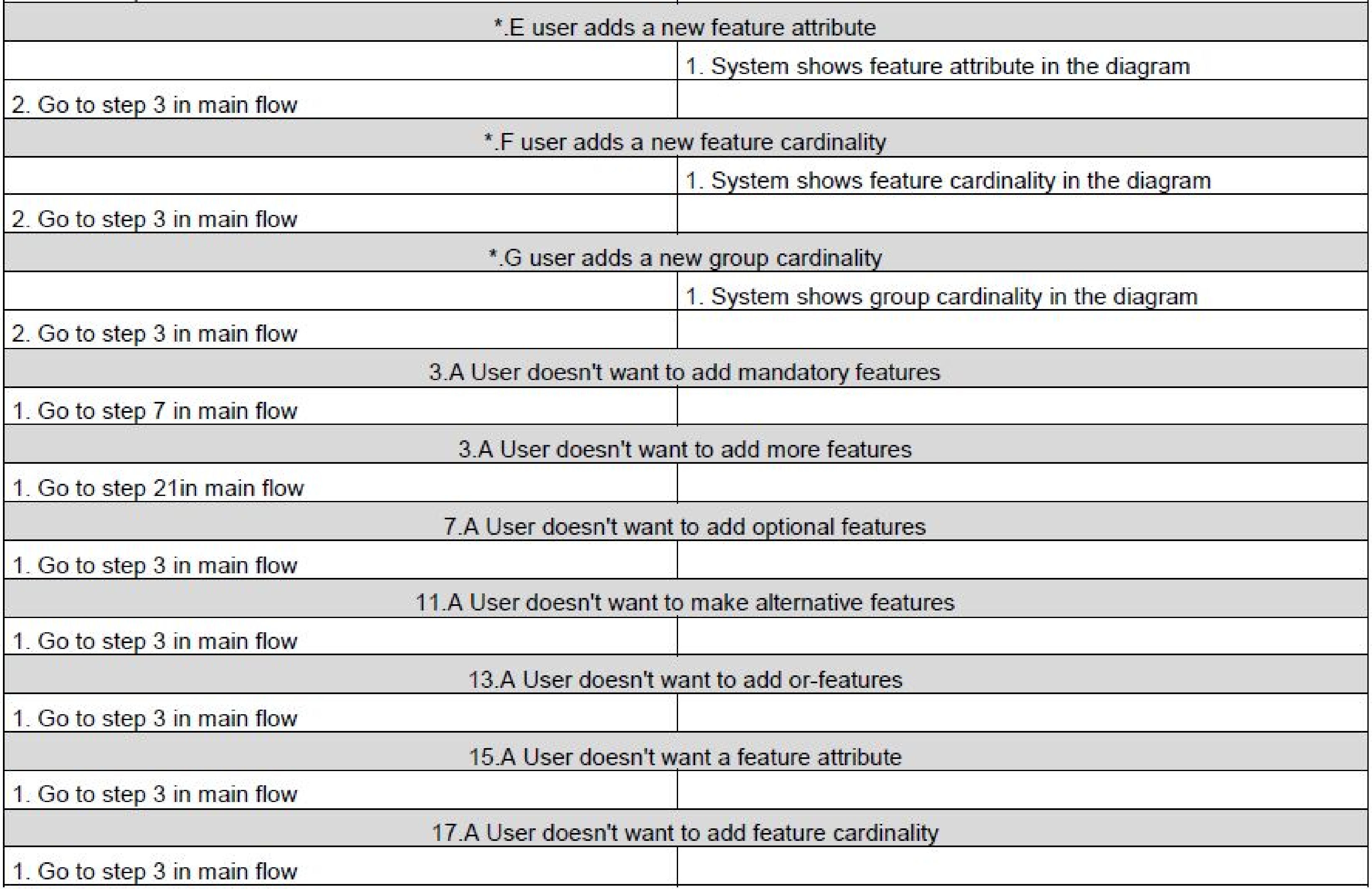


Use Case:	UC04 - Use Cardinality-based notation
Actor:	User
Precondition	User selects Cardinality-based notation
Post condition	
Actor Actions	**System Actions**
Main Flow:	
1. User adds a root feature	
	2. System shows root feature in the diagram
3. User adds a mandatory feature	
	4. System shows mandatory feature in the diagram
5. User makes the mandatory feature a child of root	
	6. System makes a relationship between parent and child features
7. User adds optional feature	
	8. System shows optional feature in the diagram
9. User makes the optional feature a child of some feature	
	10. System makes a relationship between parent and child features
11. User makes a feature an alternative feature	
	12. System makes the feature set an alternative set
13. User adds an or-feature	
	14. System shows or-feature in the diagram
15. User adds an feature attribute	
	16. System shows feature attribute in the diagram
17. User adds a feature cardinality	
	18. System shows feature cardinality in the diagram
19. User adds group cardinality	
	20. System shows feature cardinality in the diagram
21. Go to step 5 in "UC01 - Create Feature Model Diagram" main flow	



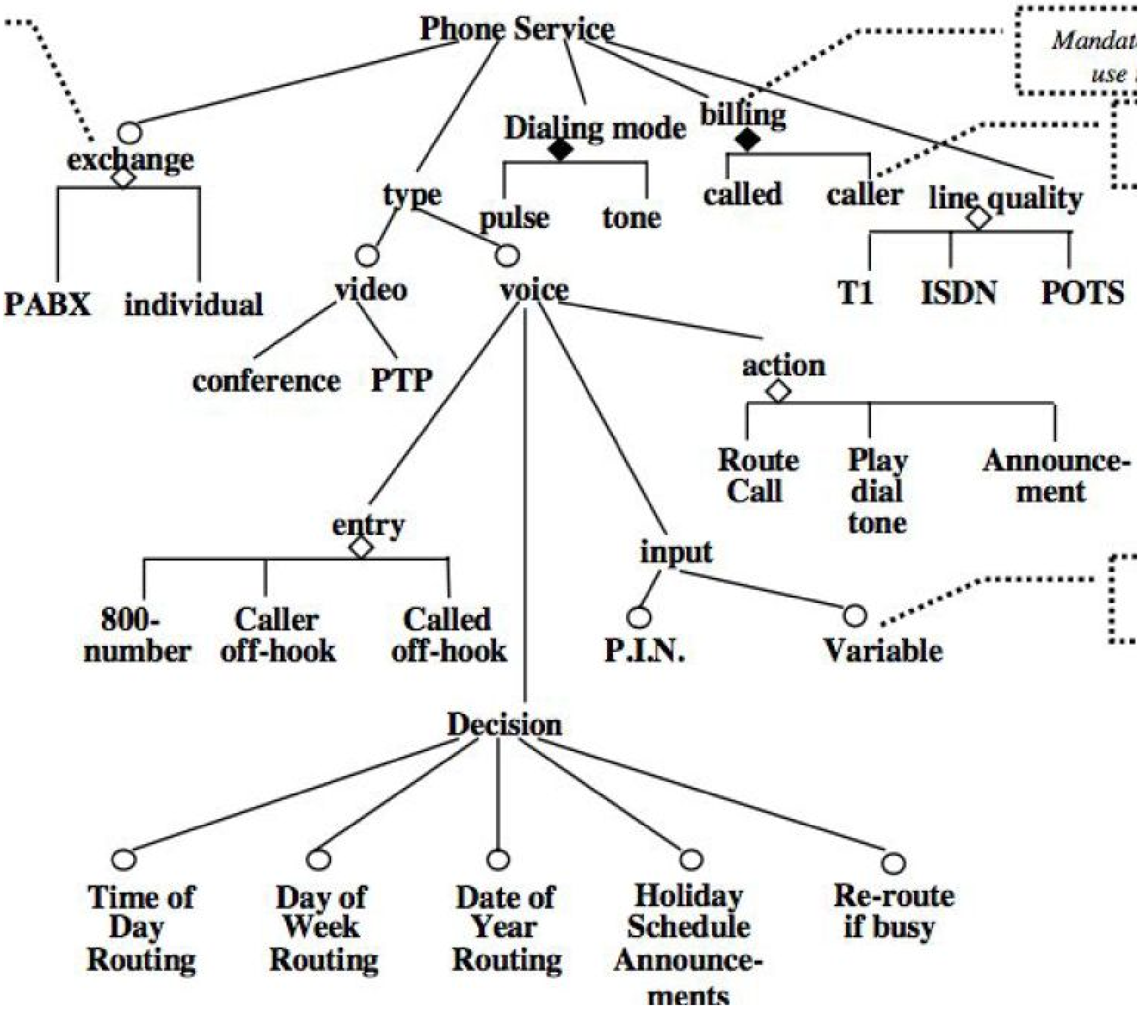
Alternative Flows:

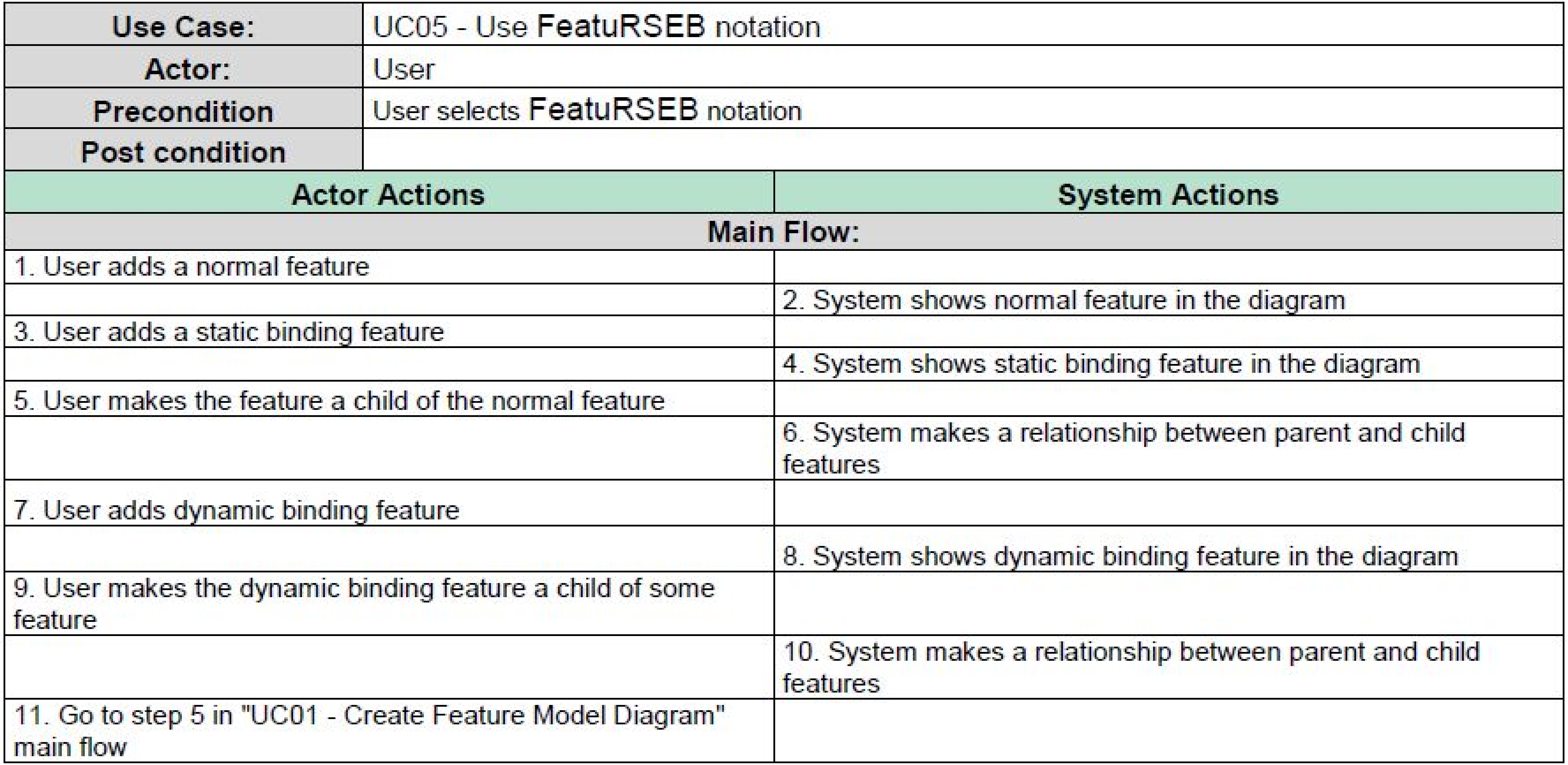
*.A user adds a new mandatory feature	
	1. System shows mandatory feature in the diagram
2. User makes the mandatory feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	
*.B user adds a new optional feature	
	1. System shows optional feature in the diagram
2. User makes the optional feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	
*.C user makes a new alternative feature	
	1. System makes the feature set an alternative set
2. Go to step 3 in main flow	
*.D user adds a new or-feature	
	1. System shows or-feature in the diagram
2. User makes the or-feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	



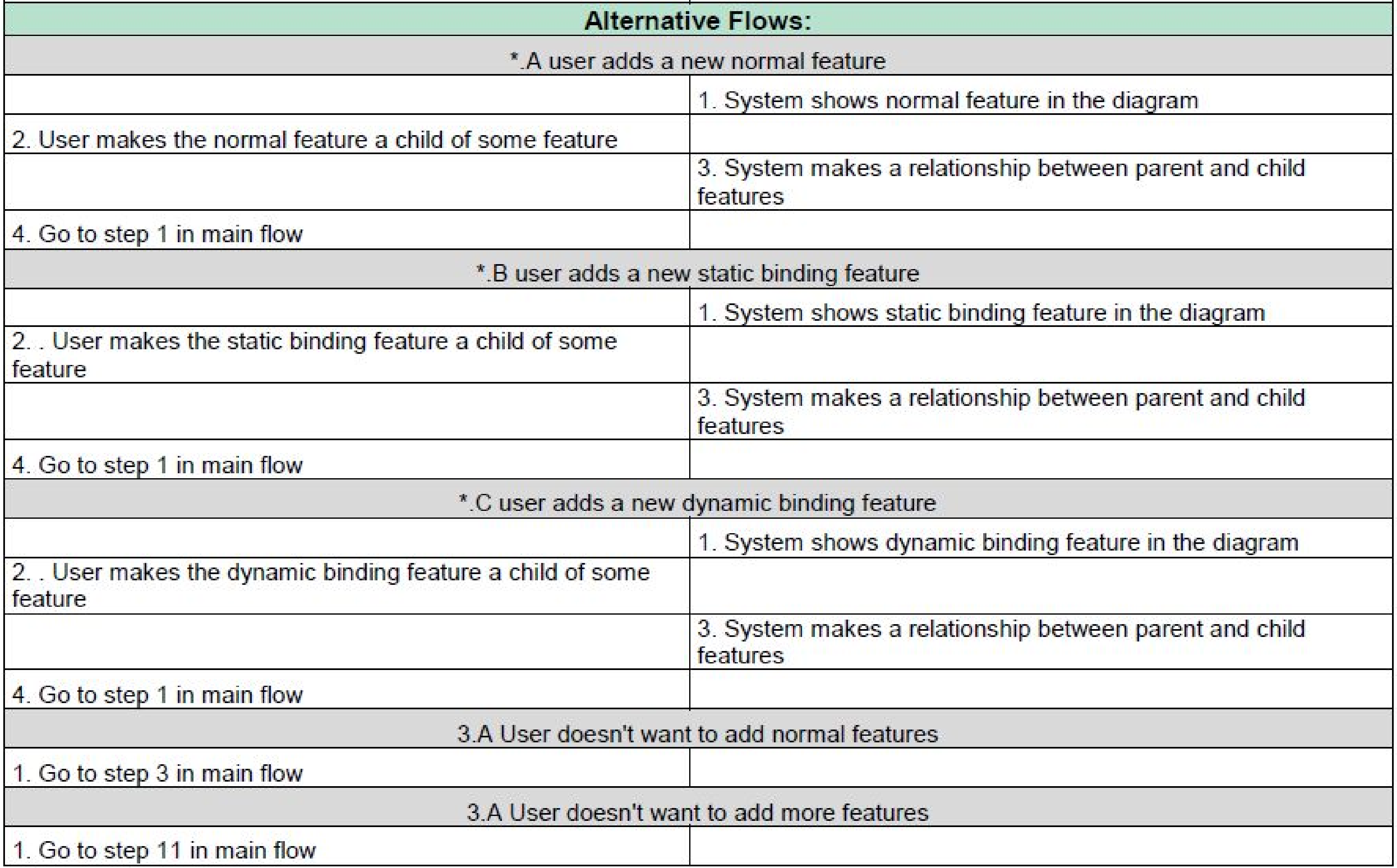
*.E user adds a new feature attribute	
	1. System shows feature attribute in the diagram
2. Go to step 3 in main flow	
*.F user adds a new feature cardinality	
	1. System shows feature cardinality in the diagram
2. Go to step 3 in main flow	
*.G user adds a new group cardinality	
	1. System shows group cardinality in the diagram
2. Go to step 3 in main flow	
3.A User doesn't want to add mandatory features	
1. Go to step 7 in main flow	
3.A User doesn't want to add more features	
1. Go to step 21in main flow	
7.A User doesn't want to add optional features	
1. Go to step 3 in main flow	
11.A User doesn't want to make alternative features	
1. Go to step 3 in main flow	
13.A User doesn't want to add or-features	
1. Go to step 3 in main flow	
15.A User doesn't want a feature attribute	
1. Go to step 3 in main flow	
17.A User doesn't want to add feature cardinality	
1. Go to step 3 in main flow	

FeatuRSEB

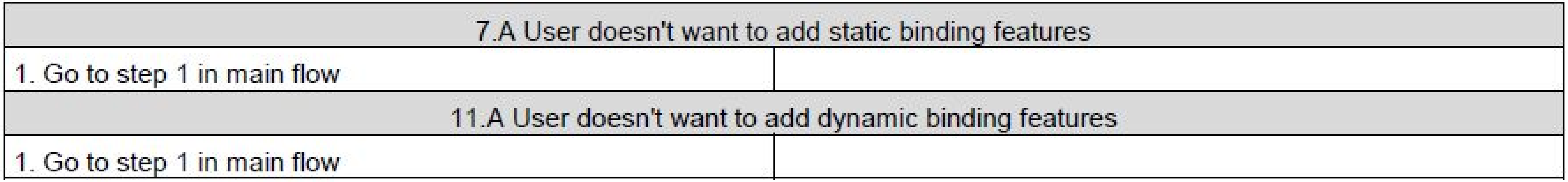

Phone Service
exchange
PABX
individual
type
video
voice
conference
PTP
Dialing mode
pulse
tone
billing
called
caller
line quality
T1
ISDN
POTS
Mandat
use
action
Route Call
Play dial tone
Announce-ment
entry
800-number
Caller off-hook
Called off-hook
input
P.I.N.
Variable
Decision
Time of Day Routing
Day of Week Routing
Date of Year Routing
Holiday Schedule Announce-ments
Re-route if busy



Use Case:	UC05 - Use FeatuRSEB notation
Actor:	User
Precondition	User selects FeatuRSEB notation
Post condition	
Actor Actions	**System Actions**
Main Flow:	
1. User adds a normal feature	
	2. System shows normal feature in the diagram
3. User adds a static binding feature	
	4. System shows static binding feature in the diagram
5. User makes the feature a child of the normal feature	
	6. System makes a relationship between parent and child features
7. User adds dynamic binding feature	
	8. System shows dynamic binding feature in the diagram
9. User makes the dynamic binding feature a child of some feature	
	10. System makes a relationship between parent and child features
11. Go to step 5 in "UC01 - Create Feature Model Diagram" main flow	

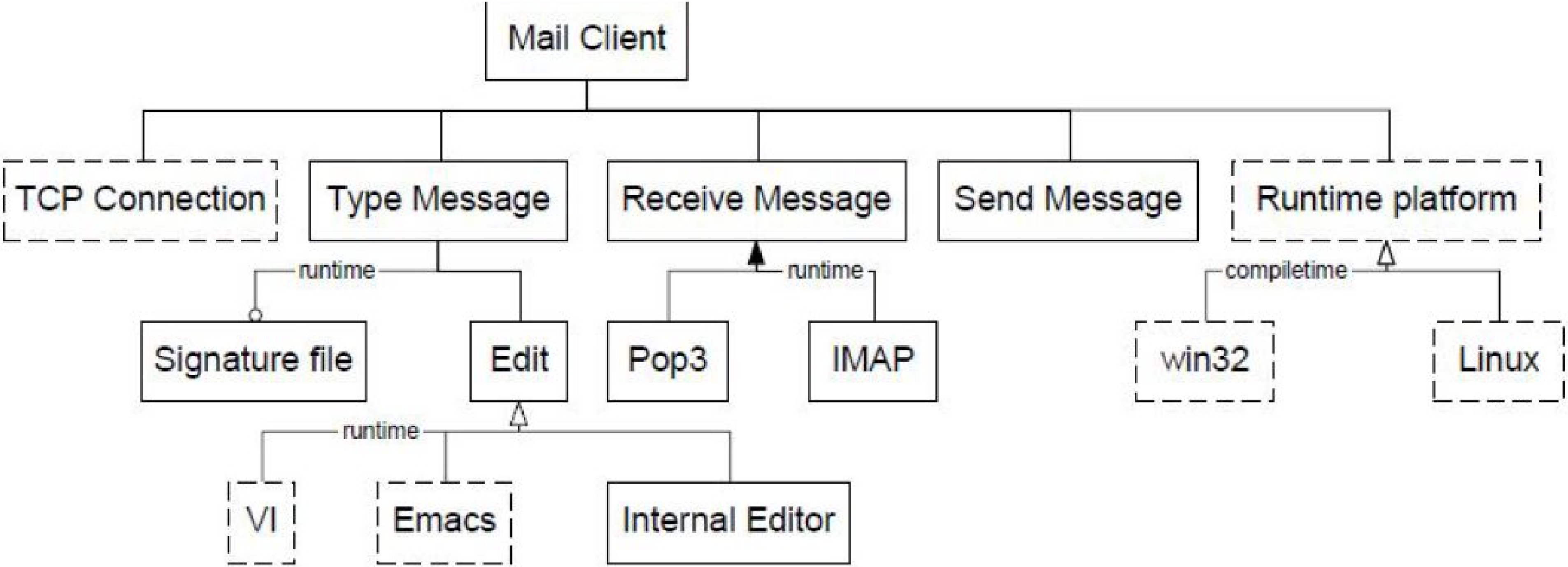


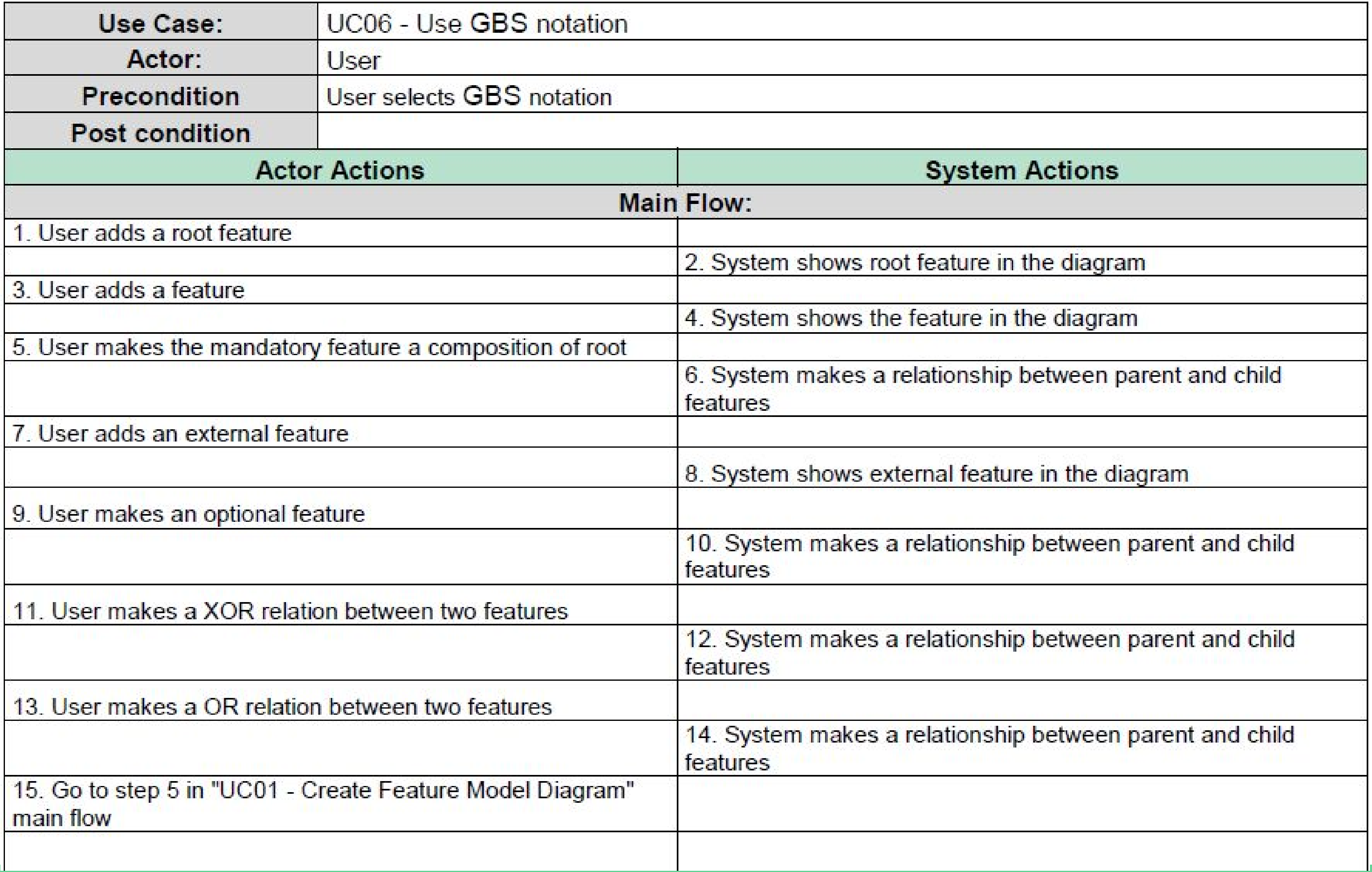
Alternative Flows:	
*.A user adds a new normal feature	
	1. System shows normal feature in the diagram
2. User makes the normal feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 1 in main flow	
*.B user adds a new static binding feature	
	1. System shows static binding feature in the diagram
2. . User makes the static binding feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 1 in main flow	
*.C user adds a new dynamic binding feature	
	1. System shows dynamic binding feature in the diagram
2. . User makes the dynamic binding feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 1 in main flow	
3.A User doesn't want to add normal features	
1. Go to step 3 in main flow	
3.A User doesn't want to add more features	
1. Go to step 11 in main flow	



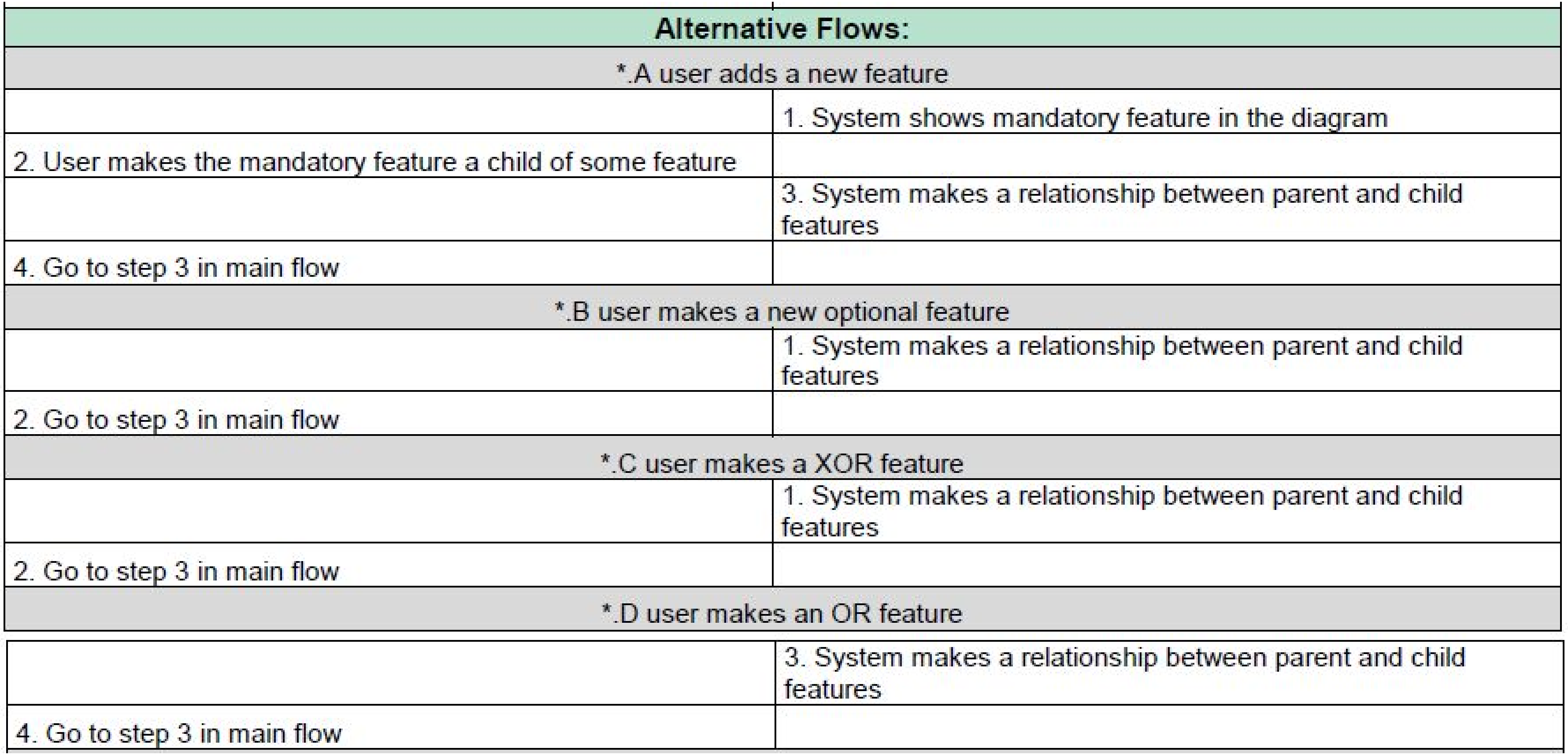
7.A User doesn't want to add static binding features	
1. Go to step 1 in main flow	
11.A User doesn't want to add dynamic binding features	
1. Go to step 1 in main flow	

GBS [van Gurp 2001]

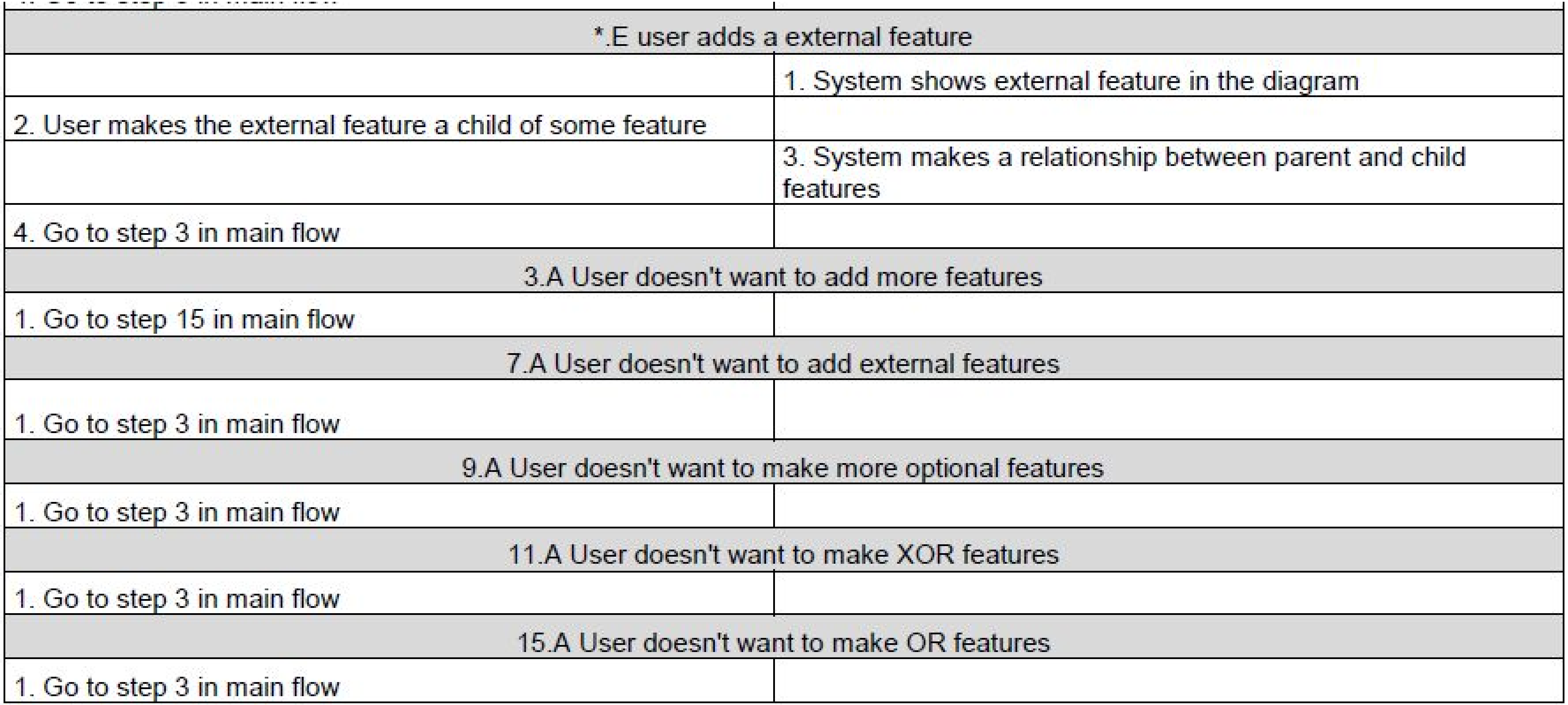




Use Case:	UC06 - Use GBS notation
Actor:	User
Precondition	User selects GBS notation
Post condition	
Actor Actions	**System Actions**
Main Flow:	
1. User adds a root feature	
	2. System shows root feature in the diagram
3. User adds a feature	
	4. System shows the feature in the diagram
5. User makes the mandatory feature a composition of root	
	6. System makes a relationship between parent and child features
7. User adds an external feature	
	8. System shows external feature in the diagram
9. User makes an optional feature	
	10. System makes a relationship between parent and child features
11. User makes a XOR relation between two features	
	12. System makes a relationship between parent and child features
13. User makes a OR relation between two features	
	14. System makes a relationship between parent and child features
15. Go to step 5 in "UC01 - Create Feature Model Diagram" main flow	



Alternative Flows:	
*.A user adds a new feature	
	1. System shows mandatory feature in the diagram
2. User makes the mandatory feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	
*.B user makes a new optional feature	
	1. System makes a relationship between parent and child features
2. Go to step 3 in main flow	
*.C user makes a XOR feature	
	1. System makes a relationship between parent and child features
2. Go to step 3 in main flow	
*.D user makes an OR feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	



4. Go to step 3 in main flow	
*.E user adds a external feature	
	1. System shows external feature in the diagram
2. User makes the external feature a child of some feature	
	3. System makes a relationship between parent and child features
4. Go to step 3 in main flow	
3.A User doesn't want to add more features	
1. Go to step 15 in main flow	
7.A User doesn't want to add external features	
1. Go to step 3 in main flow	
9.A User doesn't want to make more optional features	
1. Go to step 3 in main flow	
11.A User doesn't want to make XOR features	
1. Go to step 3 in main flow	
15.A User doesn't want to make OR features	
1. Go to step 3 in main flow	

Próxima semana?