Use Case:	UC01 - Create Feature Model Diagram		
Actor:	User		
Precondition	User is logged in		
Post condition	30		
Acto	r Actions	System Actions	
	M	ain Flow:	
1. User selects Create Diagr			
2. User selects feature mode	Inotation		
		3. System checks notation	
4. User adds features to diag	ram		
5. User saves diagram			
		6. System Validates diagram and save it in the repository	
	Altern	native Flows:	
	1.A - User se	elects Load Diagram	
		1. System shows diagrams repository	
2. User selects diagram and	clicks on load		
<u> </u>		3. System loads the selected diagram	
4. Got to step 4 in main flow			
4.A - User selected FODA notation		lected FODA notation	
T.// USCI SCIEC		System loads diagram editor for FODA notation	
2. Do use case "UC02 - Use FODA notation"			
4.B - Use selected Generative Programming notation			
		System loads diagram editor for Generative Programming notation	
2. Do use case "UC03 - Use Generative Programming		notation	
notation"			
	4 C - Use selected	Cardinality-based notation	
	4.0 000 00100100	System loads diagram editor for Cardinality-based notation	
2. Do use case "UC04 - Use	Cardinality-based notation"		
4.D - Use selected FeatuRSEB notation			
		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"		1. Cyclem loads diagram cultor for ODS flotation	
		ption Flows:	
	6.A. Dia	gram is not valid	
		1. System shows error message	
2. Go to step 4 in main flow	•		
•		•	

Use Case:	UC02 - Use FODA notation		
Actor:	User		
Precondition	User selects FODA notation		
Post condition	ost condition		
Acto	r Actions	System Actions	
	Mai	n Flow:	
User adds a root feature		2. System shave rest feature in the diagram	
3. User adds a mandatory fea	ature	2. System shows root feature in the diagram	
o. Goor adds a mandatory roc	ataro	4. System shows mandatory feature in the diagram	
5. User makes the mandatory	y 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 fe	eature a child of some feature	40 Contain makes a relationable between a great and shild	
		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 - C main flow	Create Feature Model Diagram"		
	Alterna	tive Flows:	
	*.A user adds a n	ew mandatory feature	
1. System shows mandatory feature in the diagram		System shows mandatory feature in the diagram	
2. User makes the mandatory	y feature a child of some feature		
ř		System makes a relationship between parent and child features	
4. Go to step 3 in main flow			
	*.B user adds a	new optional feature	
		System shows optional feature in the diagram	
2. User makes the optional fe	eature a child of some feature		
,		System makes a relationship between parent and child features	
4. Go to step 3 in main flow			
	*.C user makes a ı	new alternative feature	
		System makes the feature set an alternative set	
2. Go to step 3 in main flow			
2. 30 to stop o in main now	3 A ligar doesn't want	to add mandatory features	
1. Go to step 7 in main flow	J.A OBEI GOESITE WAITE	to dua mandatory roduiros	
1. Go to step 7 in main flow	0 A Hannalana H	unt to add mare factures	
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		

Use Case:	UC03 - Use Generative Programming notation		
Actor:	User		
Precondition	User selects Generative Programming notation		
Post condition			
Acto	r Actions	System Actions	
	Maii	n Flow:	
1. User adds a root feature		2. System shows root feature in the diagram	
3. User adds a mandatory fe	ature	2. System shows root roaters in the diagram	
		System shows mandatory feature in the diagram	
5. User makes the mandator	y 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 fe	eature a child of some feature		
		10. System makes a relationship between parent and child features	
11. User makes a feature an	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 - C main flow	Create Feature Model Diagram"		
	Alternative Flows:		
	*.A user adds a ne	ew mandatory feature	
		System shows mandatory feature in the diagram	
2. User makes the mandator	y feature a child of some feature		
		System makes a relationship between parent and child features	
4. Go to step 3 in main flow			
	*.B user adds a ı	new optional feature	
		System shows optional feature in the diagram	
2. User makes the optional fe	eature a child of some feature		
		System makes a relationship between parent and child features	
4. Go to step 3 in main flow			
	*.C user makes a r	new alternative feature	
		System makes the feature set an alternative set	
2. Go to step 3 in main flow			

*.D user adds a new or-feature		
	System shows or-feature in the diagram	
2. Go to step 3 in main flow		
3.A User doesn't want t	o 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		

Use Case:	UC04 - Use Cardinality-based notation	
Actor:	User	
Precondition	User selects Cardinality-based notation	
Post condition		
Acto	r Actions	System Actions
	Mai	n Flow:
1. User adds a root feature		
		2. System shows root feature in the diagram
3. User adds a mandatory fea	ature	4. Custom shave mandatom factions in the diagram
	of cations a shill of reat	4. System shows mandatory feature in the diagram
5. User makes the mandator	y reature a crilio 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 ontional for	eature a child of some feature	- construction of the state of
3. Oder makes the optional te	sactore a drilla of some reactore	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 attr	ibute	
		16. System shows feature attribute in the diagram
17. User adds a feature card	inality	
		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 - C main flow	Create Feature Model Diagram"	
Alternative Flows:		

*.A user adds a new mandatory feature		
	System shows mandatory feature in the diagram	
2. User makes the mandatory feature a child of some feature		
, and the second	System makes a relationship between parent and child features	
4. Go to step 3 in main flow		
*.B user adds a	new optional feature	
	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	
	System makes the feature set an alternative set	
2. Go to step 3 in main flow		
*.D user adds	a new or-feature	
	System shows or-feature in the diagram	
2. User makes the or-feature a child of some feature		
	System makes a relationship between parent and child features	
4. Go to step 3 in main flow		
*.E user adds a	new feature attribute	
	System shows feature attribute in the diagram	
2. Go to step 3 in main flow		
*.F user adds a new feature cardinality		
	System shows feature cardinality in the diagram	
2. Go to step 3 in main flow		
*.G user adds a r	new group cardinality	
	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 wa	ant to add more features	
1. Go to step 21in main flow		
·	it to add optional features	
1. Go to step 3 in main flow		
•	to make alternative features	
1. Go to step 3 in main flow		
·	want to add or-features	
1. Go to step 3 in main flow		
	want a feature attribute	
1. Go to step 3 in main flow		
·	at to add feature cardinality	
1. Go to step 3 in main flow	,	
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	•	

	19.A User doesn't want to add	group cardinality
1. Go to step 3 in main flow		

Use Case: UC05 - Use FeatuRSEB notation			
Actor:	User		
Precondition	User selects FeatuRSEB notation		
Post condition			
	r Actions	System Actions	
		n Flow:	
1. User adds a normal feature	9		
		System shows normal feature in the diagram	
3. User adds a static binding	feature	System shows static binding feature in the diagram	
5. User makes the feature a c	child of the normal feature	4. System shows static binding reature in the diagram	
o. Good marked the realand a c	or the normal roatare	System makes a relationship between parent and child features	
7. User adds dynamic binding	g feature		
		8. System shows dynamic binding feature in the diagram	
9. User makes the dynamic b feature	inding feature a child of some		
		10. System makes a relationship between parent and child features	
11. Go to step 5 in "UC01 - C main flow	reate Feature Model Diagram"		
	Alterna	tive Flows:	
	*.A user adds a	new normal feature	
		System shows normal feature in the diagram	
2. User makes the normal fea	ature a child of some feature		
		System makes a relationship between parent and child features	
4. Go to step 1 in main flow			
	*.B user adds a ne	w static binding feature	
		System shows static binding feature in the diagram	
2 User makes the static binding feature a child of some feature			
- Cutaro		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	
		System shows dynamic binding feature in the diagram	
2 User makes the dynamic feature	binding feature a child of some		
		System makes a relationship between parent and child features	
4. Go to step 1 in main flow			
	3.A User doesn't wa	nt to add normal features	
1. Go to step 3 in main flow			
	3.A User doesn't wa	ant to add more features	
1. Go to step 11 in main flow			
So to stop 11 in main now			

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		

1. Co to stop 1 iii maii now		
Use Case:	UC06 - Use GBS notation	
Actor:	User	
Precondition	User selects GBS notation	
Post condition		
Acto	r Actions	System Actions
	Maiı	n Flow:
1. User adds a root feature		O O otomolo a sout fort as in the Ferri
3. User adds a feature		2. System shows root feature in the diagram
3. Osci adds a leature		System shows the feature in the diagram
5. User makes the mandator	y feature a composition of root	
		6. System makes a relationship between parent and child features
7. User adds an external feat	ture	
		8. System shows external feature in the diagram
9. User makes an optional fe	ature	
		10. System makes a relationship between parent and child features
11. User makes a XOR relati	on between two features	
		12. System makes a relationship between parent and child features
13. User makes a OR relatio	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 - 0 main flow	Create Feature Model Diagram"	
	Alternat	tive Flows:
	*.A user add	s a new feature
		System shows mandatory feature in the diagram
2. User makes the mandator	y feature a child of some feature	
	,	System makes a relationship between parent and child features
4. Go to step 3 in main flow		
*.B user makes a new optional feature		
		System makes a relationship between parent and child features
2. Go to step 3 in main flow		
	*.C user make	es a XOR feature
		System makes a relationship between parent and child features
2. Go to step 3 in main flow		
*.D user makes an OR feature		
*.D user makes an OR feature		

	System makes a relationship between parent and child features	
4. Go to step 3 in main flow		
*.E user adds	a external feature	
	System shows external feature in the diagram	
2. User makes the external feature a child of some feature		
	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		

Use Case:	UC07 - Register		
Actor:	User		
Precondition			
Post condition	User is registred		
Actor Actions		System Actions	
Main Flow:			
1. User selects Register			
		2. System shows registration form	
3. User enters its data(user name, email)			
4. User selects Confirm Registration			
		5.System validates the data and confirms the registration	
Exception Flows:			
5.A. User name is already taken			
		1. System show error message	
2. Go to step 3 in main flow			
5.A. User didn't fill some required field			
		System show error message	
2. Go to step 3 in main flow			

Use Case:	UC08 - Register
Actor:	User
Precondition	User isn't logged in
Post condition	User is logged in

Actor Actions	System Actions		
Main Flow:			
1. User selects Login			
	2. System shows Login form		
3. User enters its data(user name, password)			
4. User selects Enter			
	5.System validates the data and confirms the Login		
Exception Flows:			
5.A. User name or password are not valid			
	System shows error message		
2. Go to step 3 in main flow			
5.A. User didn't fill some required field			
	System shows error message		
2. Go to step 3 in main flow			