Use Case:	UC01 - Create Feature Model Diagram		
Actor:	User		
Precondition	User is logged in		
Post condition			
Actor	r Actions	System Actions	
	Ma	in Flow:	
1. User selects Create Diagram			
2. User selects feature model	notation		
		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	
	Alterna	ative Flows:	
	1.A - User sel	ects Load Diagram	
		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 sele	ected 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			
		System loads diagram editor for Generative Programming notation	
2. Do use case "UC03 - Use Generative Programming		Totalion .	
notation"			
	4.C Use selected (Cardinality-based notation	
	4.0 030 30100104	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"		The System reader and gram curion for a cutain GEB metation.	
4.E - Use selected GBS notation 1. System loads diagram editor for GBS notation			
2. Do use case "UC06 - Use	GBS notation"	1. System loads diagram editor for GBS notation	
		tion Flows:	
	6.A. Diagram 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	ion		
Acto	r Actions	System Actions	
	Mai	n Flow:	
User adds a root feature		2. System shows root feature in the diagram	
3. User adds a mandatory fea	ature	2. System shows root feature in the diagram	
or over added a mandatory rec		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 fe	eature 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 - C main flow	reate Feature Model Diagram"		
	Alterna	tive Flows:	
	*.A user adds a n	ew mandatory feature	
		System shows mandatory feature in the diagram	
2. User makes the mandatory	/ feature a child of some feature		
2. Osci makes the mandatory leature a similar of some reature		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	
		System shows optional feature in the diagram	
2. User makes the optional fe	eature 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			
	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 13 in main flow			
so to stop to in main now			

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
	Maiı	n Flow:
User adds a root feature		2. System shows root facture in the diagram
3. User adds a mandatory fea	ature	2. System shows root feature in the diagram
		4. 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	alternative feature	
		12. System makes the feature set an alternative set
13. User adds an or-feature		
To. Good adde all of foatare		14. System shows or-feature in the diagram
15. Go to step 5 in "UC01 - Create Feature Model Diagram" main flow		14. Cystem snows of reature in the diagram
	Alternat	ive Flows:
*.A user adds a new mandatory feature		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 r	new optional feature
		System shows optional feature in the diagram
User makes the optional feature 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		
<u> </u>		

*.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	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		
Actor	r Actions	System Actions
	Mai	n Flow:
User adds a root feature		
3. User adds a mandatory fea	oturo	2. System shows root feature in the diagram
5. Oser adds a mandatory rea	ature	4. System shows mandatory feature in the diagram
5. User makes the mandatory	r 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
O Lloor makes the entional fo	esture a shild of some feeture	or eyetem eneme epitema reatare 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 cardinal	ity	
		20. System shows feature cardinality in the diagram
21. Go to step 5 in "UC01 - C main flow	reate 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	
	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	
	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		
·	t 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			
		System shows normal feature in the diagram	
3. User adds a static binding	feature	4. Custom shows static hinding feature in the diagram	
5. User makes the feature a c	child of the normal feature	4. System shows static binding feature in the diagram	
J. Oser makes the realtire at	Silina of the normal reature	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		
		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			
		System makes a relationship between parent and child features	
4. Go to step 1 in main flow	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			
Jo 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		

Use Case:	UC06 - Use GBS notation	
Actor:	User	
Precondition	User selects GBS notation	
Post condition		
Acto	r Actions	System Actions
	Mair	n Flow:
1. User adds a root feature		2. Creations also sure reset for atture in the edition record
3. User adds a feature		2. System shows root feature in the diagram
o. ocor addo a rodiaro		4. System shows the feature in the diagram
5. User makes the mandator	y feature a composition of root	
		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 relation between two features		
		14. System makes a relationship between parent and child features
15. Go to step 5 in "UC01 - C 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		

	System makes a relationship between parent and child features		
4. Go to step 3 in main flow			
*.E user adds a	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 war	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				
		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 - Do Login
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			