Visual Code Review

Group 5

Daniel Andersson Tenninge

Gustaf Carleson

Johan Björk

Patrik McKiernan

5. Design detail

5.5. Detailed Design

Class Name:	CheckCommit
Attributes:	db: A open connection to the database
Methods:	Main
Related Functional Requirement:	F.1, F.2

Method Name:	main()
Parameters:	External: SVN transaction id, SVN repository path.
Return Value:	Nonzero: Commit shall be aborted Zero: Commit shall proceed.
Description:	Inserts a changeset into the database if the code is affected by a review policy, otherwise commits to SVN server.
Data Structures:	n/a.
Pre-conditions:	n/a.
Validity Checks:	n/a.
Post-conditions:	Commit has been processed.
Called by:	External: SVN pre-commit hook i.e. a blackbox provided by SVN.
Calls:	BEDatabase, Highlight External: PySVN

Class Name:	Highlight
Attributes:	None
Methods:	highlightDiff()
Related Functional Requirement:	n/a

Method Name:	highlightDiff ()
Parameters:	Filename,oldfile(contents),newfile(contents)
Return Value:	Highlighted diff
Description:	Syntax highlights the diff between two files in HTML format.
Data Structures:	n/a.
Pre-conditions:	n/a.
Validity Checks:	n/a.
Post-conditions:	Syntax is highlighted in HTML
Called by:	CheckCommit
Calls:	External: Pygments

Class Name:	BEDatabase
Attributes:	db: A open connection to the database

Methods:	__init__
	importSQL()
	doSQL()
	newChangeSet()
	newCode()
	getPolicy()
	getAuthor()

Related Functional Requirement:	F.3

Method Name:	__init__()
Parameters:	Host, user, password, database,
Return Value:	n/a. Throws exception
Description:	Imports a .sql file
Data Structures:	db; Connection to database.
Pre-conditions:	n/a.
Validity Checks:	That we can connect to the database.
Post-conditions:	Connected to the database.
Called by:	CheckCommit
Calls:	External: Python mysql module.

Method Name:	importSQL()
Parameters:	Filename
Return Value:	n/a. Throws exception
Description:	Imports a .sql file
Data Structures:	n/a.
Pre-conditions:	n/a.
Validity Checks:	n/a
Post-conditions:	File has been executed by SQL engine.
Called by:	None
Calls:	doSQL

Method Name:	doSQL()
Parameters:	statement,arguments,cursor
Return Value:	Number of rows and row data. Depending on cursor selected. Throws exception
Description:	Executes statement, replacing any occurances of % with a argument from the argumentlist. (sql injection

	safe)
Data Structures:	n/a.
Pre-conditions:	n/a.
Validity Checks:	n/a.
Post-conditions:	SQL statement is executed.
Called by:	BEDatabase.
Calls:	External: Python mysql module.

Method Name:	newChangeSet()
Parameters:	log, author, date
Return Value:	New changeset ID.
Description:	Creates a new changeset.
Data Structures:	n/a.
Pre-conditions:	n/a.
Validity Checks:	n/a.
Post-conditions:	New changeset created
Called by:	CheckCommit
Calls:	doSQL

Method Name:	newCode()
Parameters:	changeSetId,filename,newcontent,hldiff
Return Value:	n/a.
Description:	Creates a new code entry associated with changeSetId.
Data Structures:	n/a.
Pre-conditions:	changeSetId exists in the database.
Validity Checks:	n/a
Post-conditions:	New code is associated with the changeSetId.
Called by:	CheckCommit
Calls:	doSQL

Method Name:	getPolicy(filename)
Parameters:	Filename
Return Value:	policyID.
Description:	Searches for an active policy that the filename is affected by. If more then one matching is available, it will return a valid matching, which one is not defined.
Data Structures:	n/a.
Pre-conditions:	n/a.
Validity Checks:	n/a.
Post-conditions:	A policyid is returned if available.
Called by:	CheckCommit
Calls:	doSQL

Method Name:	getAuthor(changeSetId)
Parameters:	changesetID
Return Value:	Author of that changeset.
Description:	Returns the svn-author of the changeset.
Data Structures:	n/a.
Pre-conditions:	The changeset is valid.
Validity Checks:	n/a
Post-conditions:	Author is returned.
Called by:	CheckCommit
Calls:	doSQL

Class Name:	Standard
Attributes:	

Methods:	__construct()
	check_auth()
	check_login()
	get_login()
	logout()

Related Functional Requirement:	F.11

Method Name:	__construct()
Parameters:	n/a
Return Value:	n/a
Description:	Constructor for the class Standard.
Data Structures:	n/a
Pre-conditions:	n/a
Validity Checks:	n/a
Post-conditions:	n/a
Called by:	n/a
Calls:	check_auth()

Method Name:	check_auth
Parameters:	n/a
Return Value:	Boolean, true if authenticated false otherwise.
Description:	Checks if the user is logged in.
Data Structures:	n/a
Pre-conditions:	n/a

Validity Checks:	n/a
Post-conditions:	n/a
Called by:	__construct()
Calls:	n/a

Method Name:	get_login()
Parameters:	n/a
Return Value:	n/a
Description:	Redirect the user to the login page.
Data Structures:	n/a
Pre-conditions:	n/a
Validity Checks:	n/a
Post-conditions:	n/a
Called by:	check_auth()
Calls:	n/a

Method Name:	logout()
Parameters:	n/a
Return Value:	n/a
Description:	Ends the users session.
Data Structures:	n/a
Pre-conditions:	The user is logged in.
Validity Checks:	n/a
Post-conditions:	n/a
Called by:	Standard
Calls:	n/a

Class Name:	AdminPage
Attributes:	n/a

Methods:	link_admindev()
	link_adminbra()

Related Functional Requirement:	n/a

Method Name:	link_admindev()
Parameters:	n/a
Return Value:	Link to the admin developer page.
Description:	Returns a link to the admin developer page.
Data Structures:	n/a
Pre-conditions:	Logged in with admin rights.
Validity Checks:	n/a
Post-conditions:	n/a

Called by:	AdminDev, AdminBranch
Calls:	n/a

Method Name:	link_adminbra()
Parameters:	n/a
Return Value:	Link to the admin branches page.
Description:	Returns a link to the admin branches page.
Data Structures:	n/a
Pre-conditions:	Logged in with admin rights.
Validity Checks:	n/a
Post-conditions:	n/a
Called by:	AdminDev, AdminBranch
Calls:	n/a

Class Name:	AdminDev
Attributes:	n/a

Methods:	add_developer()
	remove_developer()
	update_developer()
	reset_pwd()

Related Functional Requirement:	F.12, F.13, F.14

Method Name:	add_developer()
Parameters:	n/a
Return Value:	False if adding failed, true otherwise.
Description:	Adds a new developer to the database.
Data Structures:	USR_User.
Pre-conditions:	Logged in as administrator.
Validity Checks:	n/a
Post-conditions:	A new developer is added to the database.
Called by:	eventAddDev
Calls:	n/a

Method Name:	remove_developer()
Parameters:	developer_id
Return Value:	False if remove failed, true otherwise.
Description:	Remove a developer from the database.
Data Structures:	USR_User.

Pre-conditions:	Logged in as administrator.
Validity Checks:	Check that a developer is selected, show alert dialog if none is selected.
Post-conditions:	A developer is removed from the database.
Called by:	eventRemoveDev
Calls:	n/a

Method Name:	update_developer()
Parameters:	developer_id developer_name developer_email developer_rights
Return Value:	True if updated was successful, false otherwise.
Description:	Updates information about a developer.
Data Structures:	USR_User.
Pre-conditions:	Logged in as administrator.
Validity Checks:	Check that all fields are non-empty, if not so show an alert dialog.
Post-conditions:	Developer is updated with new data.
Called by:	eventSaveDev.
Calls:	n/a

Method Name:	reset_pwd()
Parameters:	developer_id
Return Value:	False if reset failed, true otherwise.
Description:	Reset a developers password.
Data Structures:	USR_User.
Pre-conditions:	Logged in as administrator.
Validity Checks:	Check that a developer is selected, show alert dialog if none is selected.
Post-conditions:	A developers password is reset.
Called by:	eventResetPassword
Calls:	n/a

Class Name:	AdminBranches
Attributes:	n/a

Methods:	add_branch()
	remove_branch()
	update_branch()

Related	F.13, F.14

Functional Requirement:	

Method Name:	add_branch()
Parameters:	n/a
Return Value:	False if adding failed, true otherwise.
Description:	Adds a new branch to the database.
Data Structures:	BRA_Branch
Pre-conditions:	Logged in as administrator.
Validity Checks:	n/a
Post-conditions:	A new branch is added to the database.
Called by:	eventAddBranch
Calls:	n/a

Method Name:	remove_branch()
Parameters:	branch_id
Return Value:	False if the remove failed, true otherwise.
Description:	Removes a branch from the database.
Data Structures:	BRA_Branch
Pre-conditions:	Logged in as administrator.
Validity Checks:	Check that a branch is selected, show alert dialog if none is selected.
Post-conditions:	A branch is removed from the database.
Called by:	eventRemoveBranch
Calls:	n/a

Method Name:	update_branch()
Parameters:	branch_id branch_path nr_of_reviewers rights
Return Value:	True is update was successful, false otherwise.
Description:	Updates
Data Structures:	Updates information about a branch.
Pre-conditions:	Logged in as administrator.
Validity Checks:	Check that all the fields are non-empty and show an alert dialog if not.
Post-conditions:	The branch is updated with the new data.
Called by:	eventSaveBranch
Calls:	n/a

Class Name:	Review
Attributes:	n/a

Methods:	get_reviews()
	get_review_details()
	reject_commit()
	accept_commit()
	get_source_code()
	reject_source()
	accept_source()
	send_note()

Related Functional Requirement:	F.4, F.5, F.6, F.7, F.8, F.9, F.10

Method Name:	get_reviews()
Parameters:	search_string page_nr nr_per_page
Return Value:	Data for the specified page.
Description:	Retrieves a list containing all commits up for review for the specified page based on the developers rights.
Data Structures:	CHS_Changeset
Pre-conditions:	Logged in as developer.
Validity Checks:	n/a
Post-conditions:	n/a
Called by:	eventReview eventSelectPage eventSearch
Calls:	n/a

Method Name:	get_review_details()
Parameters:	commit_id
Return Value:	Returns the details of a commit.
Description:	Gets the details of a specific commit.
Data Structures:	CHS_Changeset
Pre-conditions:	Logged in as developer.
Validity Checks:	n/a
Post-conditions:	n/a
Called by:	eventListReview
Calls:	n/a

Method Name:	reject_commit()

Parameters:	commit_id note_string
Return Value:	True if reject was successful, false otherwise.
Description:	Rejects a commit and supplies note.
Data Structures:	CHS_Changeset REV_Review
Pre-conditions:	Logged in as developer.
Validity Checks:	Checks that a note is supplied, otherwise displays an alert dialog.
Post-conditions:	The specified commits is rejected.
Called by:	eventRejectChangeset
Calls:	send_note()

Method Name:	accept_commit()
Parameters:	commit_id note_string
Return Value:	True if accept was successful, false otherwise.
Description:	Accepts a commit and supplies note.
Data Structures:	CHS_Changeset REV_Review
Pre-conditions:	Logged in as developer and no source file is rejected.
Validity Checks:	Checks that a note is supplied, otherwise displays an alert dialog.
Post-conditions:	The specified commits is accepted.
Called by:	eventAcceptChangeset
Calls:	send_note()

Method Name:	get_source_code()
Parameters:	source_code_id
Return Value:	Highlighted source code.
Description:	Returns the highlighted source code for the specified file.
Data Structures:	COD_Code
Pre-conditions:	Logged in as developer.
Validity Checks:	n/a
Post-conditions:	n/a
Called by:	eventListFile
Calls:	n/a

Method Name:	reject_source_code()
Parameters:	source_code_id note_string
Return Value:	True if reject was successful, false otherwise.
Description:	Rejects a source file in the commit and supplies note.

Data Structures:	COD_Code REV_Source_Review
Pre-conditions:	Logged in as developer.
Validity Checks:	Checks that a note is supplied, otherwise displays an alert dialog.
Post-conditions:	The specified source file is rejected.
Called by:	eventRejectFile
Calls:	n/a

Method Name:	accept_source_code()
Parameters:	source_code_id note_string
Return Value:	True if accept was successful, false otherwise.
Description:	Accepts a source file in the commit and supplies a note.
Data Structures:	COD_Code REV_Source_Review
Pre-conditions:	Logged in as developer.
Validity Checks:	Checks that a note is supplied, otherwise displays an alert dialog.
Post-conditions:	The specified source file is accepted.
Called by:	eventAcceptFile
Calls:	n/a

Method Name:	send_note()
Parameters:	message email address
Return Value:	True if mail was sent, false otherwise.
Description:	Sends an email to the address specified with the supplied note.
Data Structures:	n/a
Pre-conditions:	Logged in as developer.
Validity Checks:	n/a
Post-conditions:	Email is sent to the address with supplied message.

Called by:	accept_commit()
	reject_commit()

Calls:	n/a

Table name	Fields	Type	Foreign keys
USR_User	USR_Id	INT, PRIMARY KEY	N/A
	USR_Login	VARCHAR(10)	N/A
	USR_Password	TEXT	N/A
	USR_Name	VARCHAR(50)	N/A
	USR_Email	VARCHAR(50)	N/A

Table name	Fields	Type	Foreign keys
RGT_Right	RGT_Id	INT, PRIMARY KEY	N/A
	RGT_Name	VARCHAR(50)	N/A

Table name	Fields	Type	Foreign keys
POL_Policy	POL_Id	INT, PRIMARY KEY	N/A
	POL_Name	VARCHAR(50)	N/A
	POL_Category	VARCHAR(50)	N/A
	POL_AcceptedThreshold	INT	N/A
			N/A

Table name	Fields	Type	References foreign key
CHS_Changeset	CHS_Id	INT, PRIMARY KEY	N/A
	CHS_USR_Id	INT	N/A
	CHS_Comment	VARCHAR(1024)	USR_User(USR_Id)
	CHS_Date	DATE	N/A

Table name	Fields	Type	Foreign keys
BRA_Branch	BRA_Id	INT, PRIMARY KEY	N/A
	BRA_Name	VARCHAR(50)	N/A
	BRA_Path	VARCHAR(255)	N/A
	BRA_POL_Id	INT	POL_Policy(POL_Id)
	BRA_Active	BOOLEAN	N/A

Table name	Fields	Type	Foreign keys
COD_Code	COD_Id	INT, PRIMARY KEY	N/A
	COD_Path	VARCHAR(255)	N/A
	COD_Filename	VARCHAR(255)	N/A
	COD_CodeText	TEXT	N/A
	COD_Diff	TEXT	N/A
	COD_BRA_Id	INT	BRA_Branch(BRA_Id)
	COD_CHS_Id	INT	CHS_Changeset(CHS_Id)

Table name	Fields	Type	Foreign keys
RES_ReviewSource	RES_COD_Id	INT, PRIMARY KEY	COD_Code(COD_Id)
	RES_USR_Id	INT, PRIMARY KEY	USR_User(USR_Id)
	RES_Date	DATE	N/A
	RES_Comment	VARCHAR(1024)	N/A
	RES_Action	ENUM(‘Accept’,’Reject’), PRIMARY KEY	N/A

Table name	Fields	Type	Foreign keys
REC_ReviewChangeset	REC_CHS_Id	INT, PRIMARY KEY	COD_Code(COD_Id)
	REC_USR_Id	INT, PRIMARY KEY	USR_User(USR_Id)
	REC_Date	DATE	N/A
	REC_Comment	VARCHAR(1024)	N/A
	REC_Action	ENUM(‘Accept’,’Reject’),	N/A

		PRIMARY KEY	

Table name	Fields	Type	Foreign keys

URI_UserRights	URI_USR_Id
	URI_RGT_Id
		INT, PRIMARY KEY
		INT, PRIMARY KEY
			USR_User(USR_Id)
			RGT_Right(RGT_Id)

Table name	Fields	Type	Foreign keys

POR_PolicyRights	POR_POL_Id
	POR_RGT_Id
		INT, PRIMARY KEY
		INT, PRIMARY KEY
			POL_Policy(POL_Id)
			RGT_Right(RGT_Id)

5.6. Package diagram

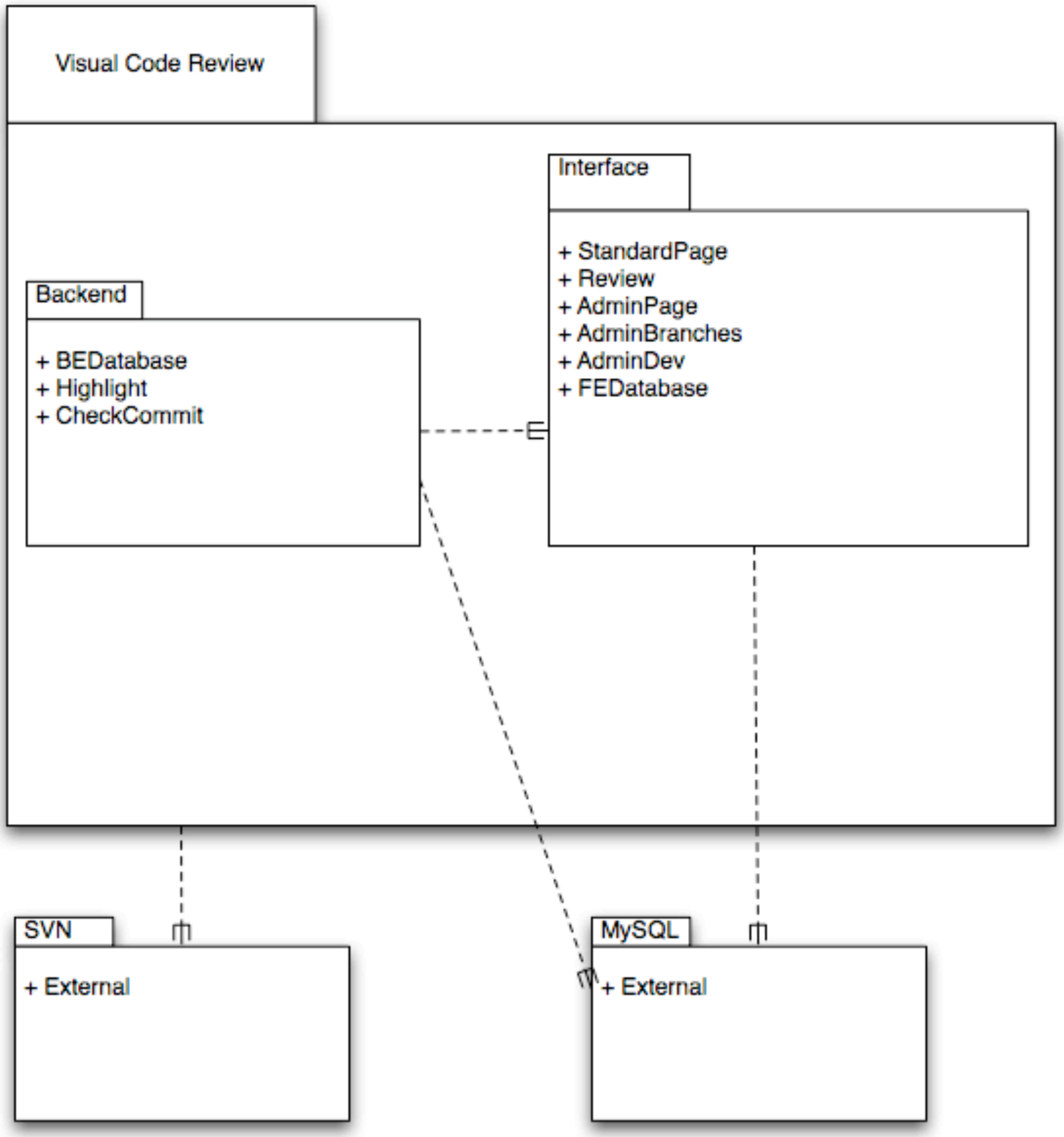

Figure 5.6.1