Bilkent University

Department of Computer Engineering

CS 453

Application Lifecycle Management

Term Project Phase 1

Team CCR:

Kasymbek	Tashbaev	21603136
Ahmet	Ay	21200701
Yryskul	Turduev	21402761
Salih Zeki	Okur	21400782
Umut	Ak	21300931

Teaching assistant: Barış Ardıç

1. BRIEF DESCRIPTION **3**

2. Criteria **3**

2.1. Availability of source code 3

2.2. Support for programming languages 3

2.3. Support for a huge number of users 3

2.4. Integration with IDE 3

2.5. Integration with other tools 4

2.6. Weakness detection functionality 4

2.7. Reporting and Statistics 4

2.8. In-code comments 4

3. Competitors Table **5**

4. Uses cases of Crucible **7**

5. Integration Diagram **12**

1. BRIEF DESCRIPTION

Crucible is a cross-platform code review tool developed by Atlassian. The installation package of Crucible also include files required for another Atlassian product **Fisheye**. Fisheye provides the ability to browse, search, and report from your repository data as it has the ability to index your entire repository history, while Crucible provides code review features. Both Crucible and Fisheye have the same architecture and highly integrated with one another. They can be used as a separate program, however, if they are used together, they run as one instance. This report discusses Crucible together with Fisheye.

2. Criteria

2.1. Availability of source code

It is important to view the source code for users to perform a code review. Therefore, the code review tool has to present a graphical and informative view of the source code repository.

2.2. Support for programming languages

Code review tool has to support all or most of the popular programming languages.

2.3. Support for a huge number of users

Code review tool may be used by big companies that have a huge number of engineers, therefore tool should support a huge number of users.

2.4. Integration with IDE

If code tool can be integrated with an IDE such as Eclipse, IntelliJ IDEA or similar, then it would much ease the work of users.

2.5. Integration with other tools

Code review tools are commonly used with the combination of other ALM tools. Therefore, if the code review tool can be integrated with many other tools, users have more flexibility in choosing the tools.

2.6. Weakness detection functionality

Code review tool should have the functionality to detect possible defects and help users to find where the detected defects are inside the code, providing a direct link, or at least the information about the line of code and the file that contains the issue.

2.7. Reporting and Statistics

Code review tool should collect data, such as the number of review requests posted by the user or the number of issues reported against each review request posted by the user, and allow users to view the statistics of data. Statistics may improve code quality using data obtained.

2.8. In-code comments

Code review tool has to provide users to comment on specific source lines, files, or an entire changeset, indicate the defects and importance of the defect.

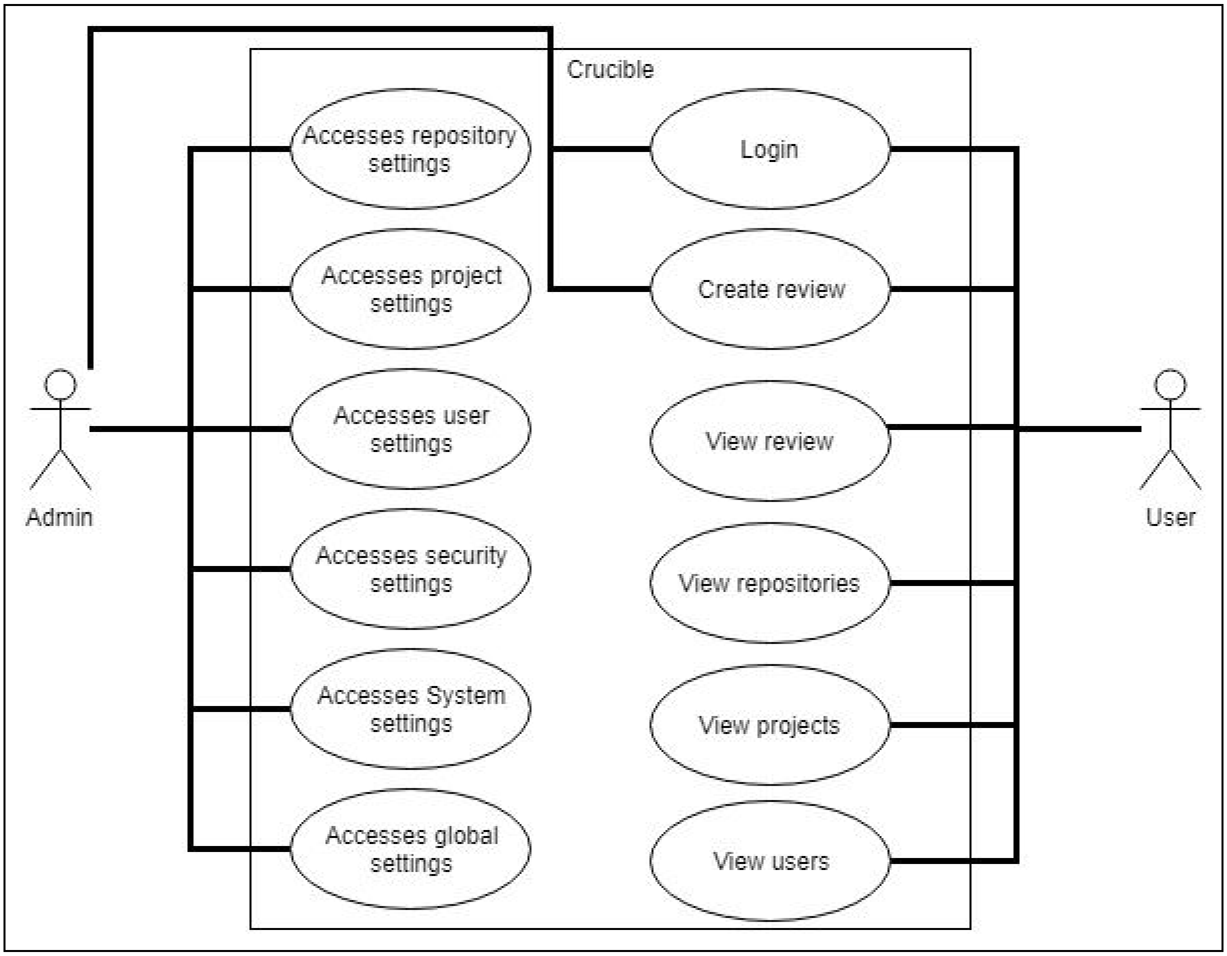
3. Competitors Table

Features / Products	**Pricing**	**Code**	**Integration**	**Inline Discussions / Comments**	**Reporting**
Crucible	Paid	Closed Source	Jira, SVN, Perforce, CVS	+	+
Collaborator	Paid	Closed Source	Git, Jira, SCMs, Eclipse, Visual Studio, Subversions, Perforce, Microsoft Word, Excel, Visio, & PowerPoint, Bitbucket, Gitlab	+	+
Review Assistant	Paid	Closed Source	Team Foundation Server, Git, Subversions, Mercurial, Perforce, Jenkins, CruiseControl.NET, MS Build	+	+
Codebrag	Free	Open Source	Git/SVN	+	-
Gerrit	Free	Closed Source	Git	+	-

Codestriker	Free	Open Source	CVS, Subversions, ClearCase, Perforce, Visual SourceSafe, Bugzilla	+	+
Rhodecode	Free/Paid	Open Source	Git, Subversion, and Mercurial	+	+
Phabricator	Free(Local)/Paid(Hosted)	Open Source	Git, SVN, Subversion, Mercurial	+	+
Veracode	Paid	Closed Source	Eclipse, IntellijIdea, Visual Studio, Jira, Github, Bugzilla, CA Agile Center, MS Team Foundation Server	+	+
Review Board	Free	Open Source	ClearCase, CVS, Perforce, Plastic	+	+

4. Uses cases of Crucible

Use case diagram-1



Uses Case Textual Description-1

Use Case Name	Accesses repository settings	Accesses project settings
Description	Manages repositories (add, remove)	Manages projects (add, remove)
Actors	Admin	Admin
Preconditions	Login as Admin	Login as Admin

Postconditions	Manages repository settings	Manages project settings

Use Case Name	Accesses user settings	Accesses system settings
Description	Manages users (add, remove)	Manages systems (backup, database)
Actors	Admin	Admin
Preconditions	Login as Admin	Login as Admin
Postconditions	Manages user settings	Manages system settings

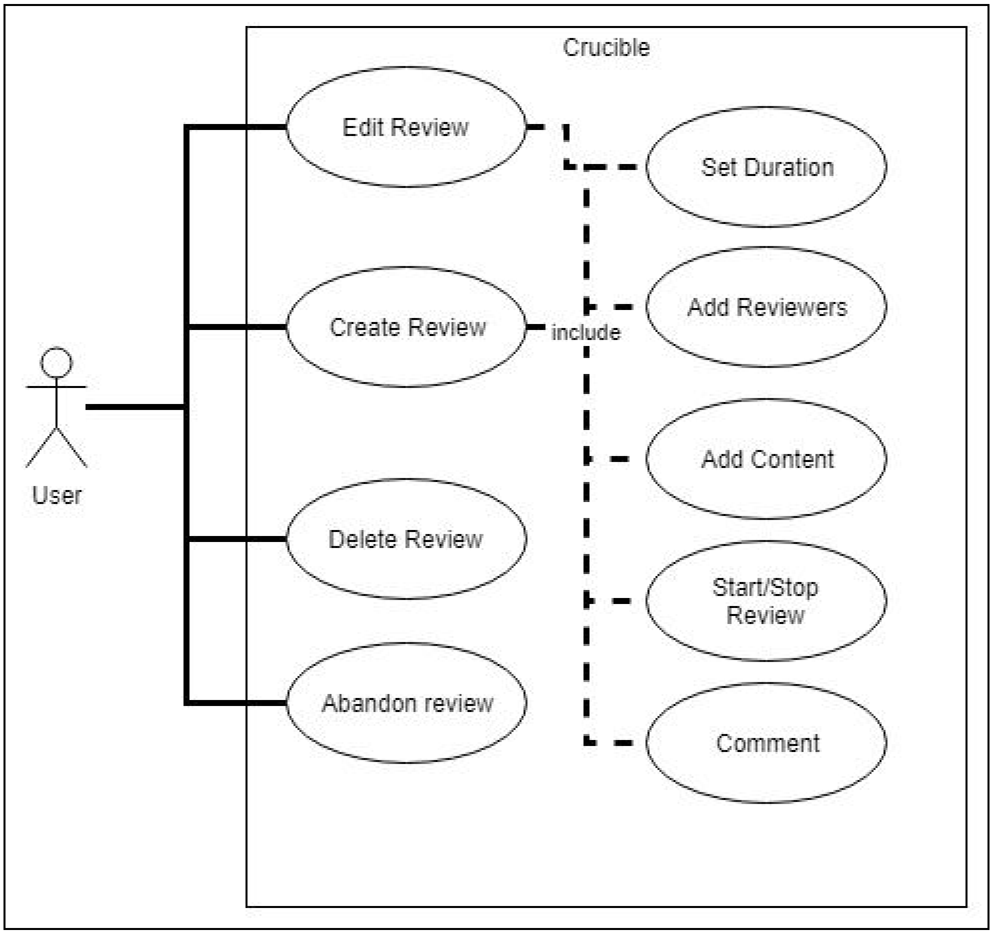
Use Case Name	Accesses security settings	Accesses global settings
Description	Manages security (permissions, authentification)	Manages global settings (Server communication, interface customization)
Actors	Admin	Admin
Preconditions	Login as Admin	Login as Admin
Postconditions	Manages security settings	Manages global settings

Use Case Name	Login	Create Review
Description	Logins to the system	Starts a new review
Actors	Admin, Users	Admin, Users
Preconditions	Having a connection to the server	Login as Admin, User
Postconditions	Logins into the system	Creates a new review

Use Case Name	View Reviews	View Repositories
Description	Views the review list	Views the repository list
Actors	Admin, Users	Admin, Users
Preconditions	Login as Admin, User. For the user, s/he should be in the reviewer list	Login as Admin, User.
Postconditions	Views the reviews	Views the repositories

Use Case Name	View Projects	View Users
Description	Views the project list	Views the user list
Actors	Admin, Users	Admin, Users
Preconditions	Login as Admin, User.	Login as Admin, User.
Postconditions	Views the projects	Views the users

Use case diagram-2



Uses Case Textual Description-2

Use Case Name	Edit Review	Create Review
Description	Edits the Review (Name, reviewers)	Starts a new review
Actors	Admin, User	Admin, User
Preconditions	Open Review Section	Login as Admin, User
Postconditions	Edits the Review	Creates a new review

Use Case Name	Delete Review	Abandon Review

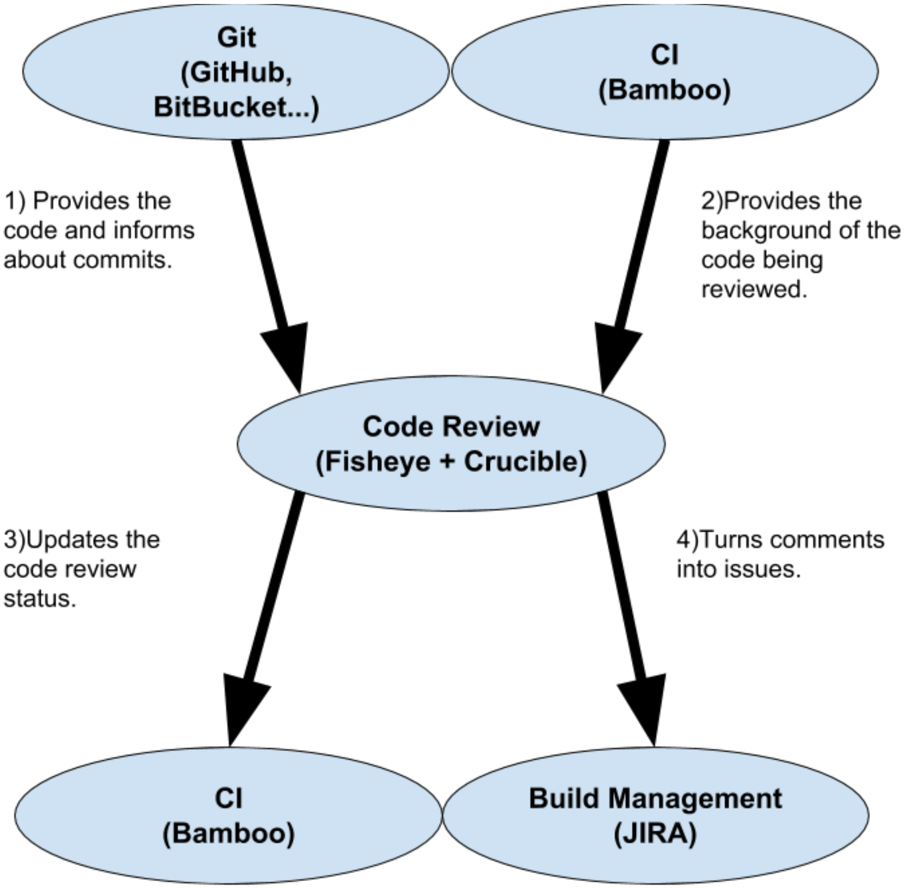
Description	Removes the review	Abandons the review
Actors	Admin, User	Admin, User
Preconditions	Open Review Section	Open Review Section
Postconditions	Removes the selected review	Abandons the selected review

Use Case Name	Set Duration	Add Reviewers
Description	Sets the duration for review	Adds the reviewers
Actors	Admin, User	Admin, User
Preconditions	Be in Edit/Create Review section	Be in Edit/Create Review section
Postconditions	Sets the duration for selected review	Add the reviewers for selected review

Use Case Name	Add Content	Start/Stop Review
Description	In this content user can search the commits or files by given filters or by users etc.	Starts/stops the review
Actors	Admin, Users	Admin, Users
Preconditions	Be in Edit/Create Review section	Be in Edit/Create Review section
Postconditions	Find needed commits etc. to review	Starts/stops the review

Use Case Name	Comment
Description	Comments on code or file
Actors	Admin, Users
Preconditions	Be in code part or file
Postconditions	Comments on code or file

5. Integration Diagram


Git
(GitHub,
BitBucket...)
CI
(Bamboo)
1) Provides the code and informs about commits.
2)Provides the background of the code being reviewed.
Code Review
(Fisheye + Crucible)
3)Updates the code review status.
4)Turns comments into issues.
CI
(Bamboo)
Build Management
(JIRA)

1) With Git integration, Crucible can reach to the code that will be reviewed. This integration would also allow crucible to inform the reviewers in case of a commit to the code.
2) Crucible's integration with continuous integration tool Bamboo increases the visibility of review task for reviewers by letting them have information about the issue that led to the creation or modification of the code, which part of the implementation the code is or what does the code used to look like before the modification.
3) Crucible also informs Bamboo about the status of the current review.
4) Comments in crucible can be turned into tasks or issues in JIRA with an integration.