

Conceptualization Document

Draft: July 03, 2012

CSFV Community Gaming Website 1.0

DRAFT: This is a draft document for review by and discussion with DARPA and the TA1 and TA2 teams. This is not a final requirements specification.

Table of Contents

1.0	Introduction	3
1.1	Overview of Application	3
1.2	Project Objectives	6
1.3	Assumptions	8
1.4	Limitations.....	10
1.5	Open Items and Risks	11
2.0	Existing Business Flow	13
2.1	Current Limitations/Problems	13
3.0	Proposed Workflow.....	14
3.1	Overview	14
3.2	Context Diagram	24
3.3	High Level Workflow	27
3.4	Use Case Diagrams.....	55
4.0	Business Requirements.....	65
4.1	Users	65
4.2	Inputs/Outputs	66
4.3	Dependencies	75
4.4	Task Requirements.....	79
4.5	Security Requirements	108
4.6	Performance Requirements.....	108
4.7	Data Migration	109

1.0 Introduction

This document is a “conceptualization” document that is intended to provide an initial description of the business requirements of the Gaming Website for the CSFV program. The purpose of this document is to provide a starting point for discussion about the requirements with DARPA and the other CSFV program performers. It is expected that DARPA and the CSFV program performers will have comments and suggested changes.

Please direct comments in advance of the kickoff to:
Ambi Del Villar adelvillar@topcoder.com

1.1 Overview of Application

Unreliable software places huge costs both on the military and the civilian economy. The current state of practice results in about one to five bugs per thousand lines of software source code. Formal program verification is the only way to be certain that a given piece of software is free of errors that could disrupt military operations in the field. Formal verification is currently done manually by specially-trained engineers. Consequently, formal verification has been too costly to apply beyond small, critical software components.

The Crowd Sourced Formal Verification (CSFV) program seeks to make formal program verification more cost-effective by reducing the skill set required for verification and by creating a public community. The approach is to transform verification into a more accessible task by creating a game that reflects the code model, is intuitively understandable, and is fun to play. Completion of the game through the community members effectively helps the program verification tool complete the corresponding formal program verification proof.

The primary research challenge faced by the entire CSFV solution is construction of automated game-level builders capable of transforming program verification models into compelling games (though that is out of scope for this conceptualization). A particular game instance is a function of the program verification tool, the property to be verified and the program being verified. Each game instance is provided to the “crowd” (public community), either via the Web or through internal domain distribution. Game solutions are used to populate a database, and then are mapped back into program annotations. Those program annotations will be sufficient to allow the program verification tool to make progress toward verifying a specific program property.

Gamification features will entertain users and attract participants to the new community – whether or not community members are experts in software validation/verification area. Games will be like original puzzles of various difficulty levels, so anyone can be involved.

The users will be excited by the games’ originality, their ability to compete with other users and obtain virtual awards, badges, and leaderboard placement. Moreover, gamers are not required to be familiar with computers and programming, and no knowledge of any programming languages or software verification principles is needed.

The source code is not directly exposed to the users through the game, instead the software will be split into small parts and converted to game levels – i.e. to some model, game task, rules, and expected goals. That conversion of the original source code will be highly automated (out of scope for this conceptualization).

The users will enjoy playing generated game levels and, while they are doing so verify specific characteristics of the software under evaluation. Results of game play from many

users will be collected by the web-site and automatically converted to the formal verification results for the related part of the software product's source code.

This conceptualization is focused on the requirements for the web site infrastructure, environment and integration of the games. It will be a web site that is available on the Internet, and also will be delivered in a form of a LiveDVD where the server and the games can be installed offline and used behind a firewall.

This document describes at a high level the infrastructure needed to implement a powerful web-portal for hosting games (related to software verification/evaluation) and supporting standard tasks for a game community (like registration, competitions, awards, interactions). The portal also hosts an automated workflow of verification and validation of software products.

The main goal of this project is to deliver a solution for efficient verification and validation of source code for large and important software products using fun, interactive games that is secure and can be developed efficiently. The solution integrates automated conversion of source code to game levels, transforming the results of played games to formal code verification output. The games will be hosted on the Gaming Website. Persisting game data and results/solutions, delivering standard community (social network) functionality, administration and reporting, management, and distribution of the Gaming Website are also parts of this project.

Please note, the games themselves, game engine, and the tools (for converting software source code to game levels and for transforming game results to formal software verification output) are required for the program but are out of scope for this document. It is assumed that the games and the conversion tools will be defined and developed by others, and that they will use interfaces that are provided by the Gaming Website.

Application opportunity

There is a need for a web-based Gaming Website for hosting games that enable formal verification and evaluation of software products' source code by non-professional users. The Gaming Website will implement a workflow for software verification/evaluation through the games. The development of this Gaming Website will make it possible for game developers to focus on developing games and related conversion tools, and then to make their games available on the Internet for users to try and enjoy, and for software owners to provide their code to the Gaming Website for formal verification.

The Gaming Website will have the following capabilities:

- C1 – an ability to create and grow a gaming community
- C2 – an ability to prepare and release game levels from the software product's source code,
- C3 – an ability to publish games and generated levels of games for playing on the web-site,
- C4 – an ability to acquire and persist game results and user's solutions,
- C5 – an ability to convert game results to the formal verification output and deliver it to the customer
- C6 – an ability to provide real-time reports and monitoring of the Gaming Website activity,
- C7 – an ability to distribute an instance of the Gaming Website with a simple installation LiveDVD,
- C8 – an ability to protect user privacy and not expose users' personal data.

C1 – gaming community creation:

The Gaming Website will have standard elements such as registration of users, user profiles, terms & conditions, help, FAQ, etc. as well as community (social network) functionality such as leader boards, high scores, awards, badges, prizes, forums, blogs, news feeds, messaging, and integration with popular social networks such as Facebook and Twitter.

C2 – game levels' preparation:

It will be possible to allow upload of software products' source code for upcoming verification. It will be possible to manage and keep game levels and conversion tools in the Gaming Website environment. It will be possible to automatically prepare a game level from the chosen part of the software product's source code.

C3 – publishing games/levels and playing games:

It will be possible to provide access to the games and the generated game levels, allowing access to those games from a PC or the mobile device. The mechanism for game delivery will depend in part on the games. It will be possible to manually administer which game levels are provided to a user and it will also be possible to use automatic adaptive scheduling to pick which game levels are delivered to a user. It will be possible to have concurrent execution of multiple games and allow many users to play same games simultaneously to allow for significantly improved verification productivity.

C4 – collecting and storing game results:

It will be possible to collect game results and user solutions achieved during game play through a web-service API and to persist that data to the database. It will be possible to get and store users' scores for played games and to use those scores for determining users' achievements and skills on the Gaming Website.

C5 – producing and delivering formal verification output:

It will be possible to automatically convert persisted game results/solutions into results that can be used for formal verification of software. It will be possible to deliver this output to the appropriate customer by a communication mechanism such as e-mail, or through an API.

C6 – monitoring and real-time reporting of Gaming Website activity:

It will be possible to provide "live" statistics on the usage of Gaming Website, hosted and played games, game results and progress, user actions, verification results, and so on. The results can be dynamically displayed in graphical and/or textual form.

C7 – distributing and installing the Gaming Website:

It will be possible to generate a LiveDVD – i.e. a disk for distribution of the Gaming Website for installation at a customer site. It can be an offline installation (without an access to the Internet and to the public Gaming Website), or online installation (with access to the public Gaming Website through Internet). The customer will be able to use that installation for his/her own private management and verification of software products.

C8 – protecting privacy and users' personal data:

It will be possible to protect user personal data by not exposing it through the application and games, by allowing some anonymous game play, and by minimizing collection and use of personal data. Additional security can be achieved through standard logging and auditing functionality.

Additional (common) opportunities:

The web site should have good performance and security. It will be possible to perform regular administrative and management actions on user accounts, games' options, and configurations.

Definitions

Acronym, statement	Description
API	Application Programming Interface
CD	Compact Disc
CMS	Content Management System
CSFV	Crowd Sourced Formal Verification
CSV	Comma-Separated Values
DVD	Digital Versatile Disc
FAQ	Frequently Asked Questions
FTP	File Transfer Protocol
Gaming Website	A web-server that will host and deliver game levels to users, capture game results and found solutions. It produces all the infrastructure and environment for formal verification of the software by community.
GUI	Graphical User Interface
HTML	HyperText Markup Language
ID	Identification
JSON	JavaScript Object Notation
LiveCD, LiveDVD	A special disk allowing distribution of the Gaming Website to a client and installation on the client's site. It will have the operating system, web-server, and web applications allowing software verification through the game workflow on the client's web-site.
MS	Microsoft
OAuth	Open Authorization – an open protocol for user authorization on multiple web-sites.
OpenID	Open Identification – an open protocol for user authentication on multiple web-sites.
OS	Operating System
PC	Personal Computer
PHP	“PHP: Hypertext Preprocessor” – a recursive description, previously known as “Personal Home Page”.
RESTful	Representational State Transfer Web Services
ROI	Return On Investment
TC	TopCoder
TXT	A plain text file format
XML	eXtensible Markup Language

1.2 Project Objectives

The business objectives of this project are listed below. Delivering these objectives will deliver the expected benefits of the application.
Must have:

- General Objectives:
 - To implement a web-based application for hosting externally developed online games allowing formal verification and evaluation of software products by public gaming community.
 - To provide general infrastructure of a Gaming Website (hosting, registration, user profiles, terms & conditions, web-site usage, help, FAQ, news, etc.).
 - To provide standard community (social network) functionality for users' conversations – like leaderboards, high scores, awards, badges, prizes, forums, wiki, blogs, user walls, discussion on most popular social networks (like Facebook, Twitter), etc.
 - To implement flexible and powerful web services API for accessing general features, game, administration, and reporting functionality; and for getting/receiving all the related data from the Gaming Website.
 - To achieve hide source code from game players or even from game administrators.
 - To protect user privacy by not exposing personal information through the application, by allowing some anonymous usage of the application and by minimizing of personal data required by the application from the user.
 - To integrate authentication/authorization with OpenID and OAuth protocols.
 - To achieve good performance of the web-application even for a large count of users.
 - To send helpful e-mail notifications to users upon the occurrence of important events during the application workflow.
 - To support logging of errors/warnings/exceptions, auditing all the user actions and data modifications, tracking all the analytics and statistics data during the application execution (in all the modules).
 - To support GUI accessibility according to Section 508 standards: http://www.access-board.gov/sec508/standards.htm#Subpart_b
 - To localize GUI and support multiple languages.
- Gaming Objectives:
 - To allow verifying software products through interesting games.
 - To integrate externally developed games for verification of software products in interesting, user friendly, and visually appealing manner.
 - To allow users who do not have a background in formal verification or software to play the games, so to NOT require professional formal verification skills and abilities.
 - To allow manual and adaptive assignment of game levels to game players according to their skills and experience.
 - To integrate adaptive scheduling of games, game levels and scaling of software verification by multiple users having required experience and skills.
 - To publish games on the web-site and allow accessing, running and playing those games on the PC or the mobile device.
 - To support short contests and longer tournaments on the published games – so game players will be constantly engaged by the application and there will be a lot of returning users for the Gaming Website.
 - To maximize verification productivity through attracting many new people and keeping existing users for a long time.
 - To achieve throughput of software verification through concurrent execution of multiple games and allowing many users to play games simultaneously.
 - To automatically calculate and persist user scores and to use those scores for determining users achievements and skills on the Gaming Website.

- To properly determine when results are complete and ready to be returned to the software program owner.
- To support a flexible and configurable workflow of software verification process.
- Administration Objectives:
 - To allow both online and offline delivery of software source code to the Gaming Website for further verification by public gaming community.
 - To integrate into the Gaming Website externally developed conversion/verification tools (for generating game levels and converting game results into formal verification results).
 - To convert source code of software products to be verified into game levels with externally developed conversion tools.
 - To implement administrative and management actions on user accounts, content moderation, games' options, etc.
 - To allow easy configuration of the application, games and selecting of the needed external converting/verification tools and compilers.
- Reporting Objectives:
 - To automatically acquire game results and user solutions achieved during game play; and to persist that data to the database.
 - To automatically generate formal software verification output and various reports (in various formats) according to the game results and user optimal solutions achieved during the game play.
 - To deliver formal verification output to the related customer (software owner) either in online or offline mode.
 - To track application usage data, game performance, statistics, users' activity, and produce analytical reports on that data.
 - To provide various statistical and analytical reports on Gaming Website usage, users' activity, and software products' verification process.
 - To support full customization of generated reports.
 - To implement real-time reports with "live" statistics and analytics information on the Gaming Website usage, users' activities, and verification process progress.
 - To support automated scheduled generating of reports and sending them to FTP server.
 - To support exporting reports in various data formats and printing reports.
- Build and Installation Objectives:
 - To allow easy and flexible build of a LiveDVD – i.e. a disk for distribution of the Gaming Website for installation at a customer's site.
 - To support easy data transfer between public and private instances of the Gaming Website (either in online or offline mode).

1.3 Assumptions

Some assumptions generally must be made in order to write a succinct definition of the application. Some assumptions are technical (e.g. "the new system will employ a normalized database schema"), while others are business natured (e.g. "the client will provide an enterprise Oracle environment"). Assumptions critical to the success of this project are listed below:

- Architectural assumptions:
 - The Gaming Website will be web-based and publicly available on the Internet.
 - The games will be fun to play.
 - It will be possible for the game results to be useful in the formal verification of software.

- The games will be compatible with and integrated into the Gaming Website architecture.
- The application will be brand new, there is no existing system.
- Some existing approaches from nonamesite.com can be re-used (like authentication/authorization, CMS, blogs, forums, etc. community-related and scoring functionality).
- The Gaming Website will communicate with the games through the Web Services APIs to be developed.
- Users can be from different time zones all over the world, so at least some localization of the application is required.
- The database structure has to be flexible and generic, so all games can simply expand it. Data schema design for new games has to be a well defined task in the actual game development.
- There will be a common Game Data model defined in the Game Web services API architecture. This has to be uniform and standardized, and individual games may have their own database and structure separate independent of the common Game Data model – and this will be out of scope.
- The game designers will not be constrained to any particular data model for their games, and will not need to be aware of the common Game Data model used by the website. Rather they will be required to conform to the defined interface.
- The application will be scalable, so the same game level can be simultaneously played by multiple users.
- There are no special GUI needs, but GUI accessibility has to be implemented according to Section 508 standards: http://www.access-board.gov/sec508/standards.htm#Subpart_b
- There are 5 modules, defined by the client for the application:
 - M1 – Gaming Community Website,
 - M2 – Gaming,
 - M3 – Admin Tools,
 - M4 – Reports,
 - M5 – Build and Installation DVD.
- Infrastructure assumptions:
 - The software products to be tested are provided as a source code.
 - The application will be like a game web-site and it will support standard infrastructure for the public gaming community.
 - The application can be delivered to customers on LiveDVD for installing private instances of the Gaming Website.
 - Users will access the application through a web-browser from the PC or mobile device.
 - User's PC can be an MS Windows-based, Linux machine, or Apple Mac.
 - Application users will have an Internet access.
 - Mobile device types to be supported are not defined yet.
 - The tools and game levels will be externally developed (by Game Developers, out of scope) and considered as “black boxes” in this application.
 - This application itself is not military, but it is expected to be attacked by hackers on the Internet and so must be resistant to such attacks.
 - The application has to achieve high security and performance.
 - OAuth and OpenID will be integrated for authorization and authentication in the application.
 - Access to the Web Services API will be authorized through OAuth.
 - User privacy will be paramount – games will know only about user IDs and other personalization information required for the games; the

application will require at a minimum e-mail, username/password, and date of birth information from the user.

- Functionality assumptions:
 - The application will perform a formal workflow for software product verification.
 - There can be private installations of the Gaming Website accessible only by the limited set of customer's staff and that are independent of a public Gaming Website.
 - At least one Site Administrator has to present in the system. The last Site Administrator cannot be removed.
 - There will be Game Administrators who can access game level converting and verification Tools to be provided. These tools will be used to create/convert software source code to game levels.
 - Some game levels can be predefined, others – dynamically created by the application and the adaptive scheduling sub-system.
 - Adaptive scheduling provides intelligent assignment of game levels to the users according to their skills, experience and performance.
 - Source code will not be directly exposed to game players.
 - Source code converting will be highly automated.
 - Game levels will be generated on parts of the sources code of the software products. Generation of game levels will be performed by special tools, provided by Game Administrators.
 - Software products to be verified can be small, medium or large – there are not constraints on their size.
 - Game players are not required to know anything about software development or verification. They just need to learn the objective and rules of the specific game or game level, and to achieve the specific goals of the game and game levels.
 - Reports will be displayed and exported in HTML, CSV, XML, JSON. It will be also possible to export reports in TXT and PDF formats.
 - There will be also real-time reports providing "live" statistics, analytics and software verification results data.
 - Reports can contain graphs and charts.
- Technical assumptions:
 - Content management can be performed through some CMS (like Liferay, Joomla, Drupal, Magnolia) – they will be used in Proof of Concepts to compare performance and reliability. Automated volume tests and benchmark performance evaluated will be applied to all those systems to determine the best one for further usage in the application.
 - Java or PHP-based technologies are preferred for this application.
 - MySQL will be used as the database for the application.
 - A single public instance of the application is enough – it will be placed on some server or on a cluster of servers.
 - Client provided documentation (modules diagram, overview, objectives and assumptions were re-used an improved).
 - Linux platform is preferred for Gaming Website web-site hosting and LiveDVD creation.
 - Web Services have to be RESTful and secured.

1.4 Limitations

Every project has limitations on the scope of the problem it attempts to solve, on the capabilities of the implementation technology, etc. Some limitations are assumptions on the extent of feature scope. Others are restrictions on resources or methods for achieving the objectives. The limitations of this project are listed below:

- Games, game engine, game logic and dynamics, test games (proof of the concept), conversion and verification tools (and compilers) will be externally implemented (by Game Developers). Their functionality is out of scope for this conceptualization. The games will need to be compatible with the Gaming Website architecture and APIs. The Game Developers will provide the game and tool installation artifacts for the LiveDVD.
- Adaptive scheduling of games and game levels will be implemented externally. That functionality is out of scope for this conceptualization.
- There are no requirements on any special peripheral devices except a printer.
- Designing of a database schema is not in scope of this conceptualization.
- Evaluation of game results will be performed by the game infrastructure tools and is out of scope for this conceptualization.
- Presence of source code is required for the software product under the test in the initial release of this project.
- The LiveDVD installation will need access to data storage.

1.5 Open Items and Risks

During the conceptualization of the project and in the process of writing this document, some issues may have been discovered, and others may remain open.

Issues for the Next Phases of the Application

- There is a risk that automatically generated game levels can be too difficult, boring, or not interesting.
- There is a risk that generated game levels can significantly expose important details of the original source code of the software products being verified.
- There is a risk that not enough game results will be collected during game play, or some results will be of a low quality, so the conversion of that data to the formal verification output will be incomplete or contain mistakes.
- There will be a lot of concurrent users, so the Architect has to elaborate on thread-safety issues: no race conditions and no data should be corrupted by simultaneous access to the system from many users.
- There is a risk that network connection with Web Services can be lost, so programmatic access to the application, games and tools integration will not be available at that moment.
- A lot of notification e-mails can be sent by the system. It can be a performance bottleneck and, may be, recognized as a spam by some e-mail spam filters. The Architect has to elaborate on this issue.
- There is a risk that counts of unique users are not explicitly defined yet per each user role.
- There is a risk that counts of concurrent users are not explicitly defined yet per each user role.
- There is a risk that types of supported mobile devices are not defined yet.
- There is a risk that cost of web-site hosting, maintenance, game creation and processing can be quite high and comparable with manual verification of the software source code by professional staff if the size of public gaming community is too small.
- There is a risk of hacker attacks on the application and Gaming Website.
- There is an open issue – to implement all the needed games, game engine, converting/verification tools and compilers. It is up-to the Game Developers to implement and to coordinate compatibility with the Gaming Website architecture.
- There is an open issue on how to prepare game levels – whether as predetermined levels from the source code, or dynamically created from the source code (up-to the Game Developers).

- There is an open issue – to implement adaptive scheduling of games and game levels according to the users' skills. It is up-to the Adaptive Scheduling developers.
- There is an open issue – to define the database schema for entire set of the data, for games, tools, and game engine. It is up-to the upcoming competitions with input and feedback from the Game Developers.
- There is an open issue – to define all the Web Services APIs (for General, Game, Admin Tools, and Reporting modules). It is up-to the upcoming competitions.
- There is an open issue – to define the API between the Gaming Website and native mobile games. It is up-to the upcoming competitions.
- There is an open issue – to define the layout, formatting, and detailed structure of reports. It is up-to the upcoming competitions.
- There is an open issue to define the performance of the CMS – some upcoming contests will determine the most fast and reliable CMS for this application.
- There is an open issue that there are no specifics on the software to be verified. All details will be defined by the Game Developers.
- There is an open issue to define a test game as a Proof of Concept – to check the application and Web Services API performance.

2.0 Existing Business Flow

This is a new system and there is no need to describe the existing application according to the manager's clarification on the forum.

2.1 Current Limitations/Problems

Limitations of and problems with the current application are listed below. These are the issues that the new application strives to address.

- The previous business process of software source code verification was performed fully manually with very specific tools.
- Previously, software verification was done by skilled professional personnel.
- Previous business process did not scale well – it was too costly to perform software verification for large projects. Only small and important software components could be verified.
- Source code was previously exposed to verification personnel, which presented security issues and further limited the personnel who could work on the code.

3.0 Proposed Workflow

3.1 Overview

This section includes a concise description of the features and operation of the proposed solution. It also compares and contrasts the solution with the current process. The context and workflow of the proposed solution are defined in subsequent sections.

Comparison with an Old Approach

There was no previous system, the previous business process was fully manual, and not many details of an old solution were provided by the client. So, we can compare only high level key points of the old and new business processes.

The previous business process was fully manual, required highly skilled and very costly personnel. It was based on special tools, did not produce automatic reports on software verification results. The old business process took too much time, did not scale (was very costly for large projects), was quite error-prone. There were security concerns, because the source code of the software was exposed to verification personnel.

- *In contrast, the new business process will be mostly automated – there will be a special web-based environment with a Gaming Website performing an automated software verification workflow, data persistence, verification tools, compilers, game level generators, reporting, and social network functionality for a public community. The automated game-based workflow will automatically convert software source code to game levels, publish games on the web and provide game play, collect results and convert them to formal verification reports.*
- *The new business process will be based on non-professional users from the created public community. They will play logical games and puzzles to verify software source code in an interesting and visually appealing manner. The new application will be based on standard web technologies. Community members will not be required to have any special skills, and it is expected they will perform the verification cheaply (or for free).*
- *The new business process will scale very well by attracting a large count of community members and parallelizing their work. Cost will be reduced if community members are willing to perform verification because it is fun. A large number of community members will also significantly reduce verification time and greatly speed-up the new business process.*
- *There will be automatic collecting of games' results, converting and presenting them as formal verification results. The number of verification errors will be reduced in the new application due to automation.*
- *Security will be improved in the new business process because the source code of the tested software will not be provided to community members.*

Handling of Capabilities C1 – C8 in the Workflow

Capability C1 (gaming community creation) can be provided by using an off-the-shelf CMS and existing components. The application will be hosted on a dedicated web-server and provide a homepage (like a landing page), which implements an exciting and visually attractive start to software verification gaming world. The user will find information about the application (help, FAQ, terms & conditions, rules, contacts), news; read-only access to forums, blogs, posted games, contests/tournaments, ability to print any web-page and play some example games without any registration – as an anonymous user. The user can register to the application to get additional functionality.

The registered user will be able to manage his/her profile and determine what parts of the data to expose to other users. He/she will have a useful dashboard with the most important and useful data and features of the application. The registered user can

communicate with other registered users (through blogs, forums, social network integration) about games and other topics. Game players can achieve awards, badges, and other yet to be determined rewards based on his/her performance on the games. Registered users can freely view all the leader boards, scoring on each game and game level, awards, virtual badges and prizes of other users.

Capability C2 (preparation of game levels from the software product's source code) is achieved by the Game Administrator, after the Software Owner (of a software product to be tested) provides the source code of that product to the Gaming Website. The Game Administrator will upload source code of the software program to be tested to the application and select an appropriate existing game level generator on the Gaming Website. All the game level generator tools will be implemented externally and provided to the application by the Game Administrator. He/she will generate game levels by the chosen external tool and store the generated game levels to the application.

Software Owners will be also able to convert their software source code into game levels. This can be useful, for example, if they prefer to not expose source code even to Game Administrators.

The Game Administrator can manually determine which game levels to play by which users on the Gaming Website. Some game levels can be set as restricted, hidden or postponed, but others – available for playing by game players. Then, the released game levels can be adaptively scheduled for game players according to users' skills.

Please note, the application will provide full integration of the game level generation tools, so those tools can be implemented later and easily inserted to the CSFV project.

Capability C3 (publishing games/levels and playing games) is achieved by the Gaming Website application. The games itself are implemented externally (by Game Developers) and provided to the application by the Game Administrator. He/she can configure options of the installed games and those games are published on the Gaming Website. The Game Player can play them according to chosen or automatically scheduled game levels. The games itself are out of scope, but the gaming infrastructure will be provided by the application, so games can be easily integrated in the future. The user can play one or multiple levels of one or multiple games.

Capability C4 (receiving and persisting game results and user's solutions) is mostly performed by the System and is exposed through the special Game Web Services API. The Game Administrator has to provide needed external tool for converting game results and user solutions to formal software verification output. The games will provide the game results, user solutions and intermediate game data to the application through the API. The system will persist the game data to the database (a generic database schema may be used for data storage and integration with any game). The users' achievements and scores will be stored as well. The system will store game level completion status when the related game determines that the game level is completed. The persisted data will be used for generating formal verification output.

Capability C5 (producing and delivering formal verification output) is configured by the Game Administrator and then automatically performed. The Game Administrator can manually map completed game results to formal verification output (like documents) by selecting the needed verification tool (all tools are out of scope for this conceptualization). The chosen external tool will convert collected and stored game results and acquired user solution to the software product verification and validation data output.

Generated formal verification output will be provided to the Software Owner (either online – through some Reporting Web Services API, or in the offline mode). The Software

Owner can manually download the generated formal verification results from the Gaming Website, or use an API.

There will be software verification reports in various formats – HTML, CSV, XML, and JSON. The application will be able to display then to the user, allow to export them as a file in various formats (including PDF), or easily print reports.

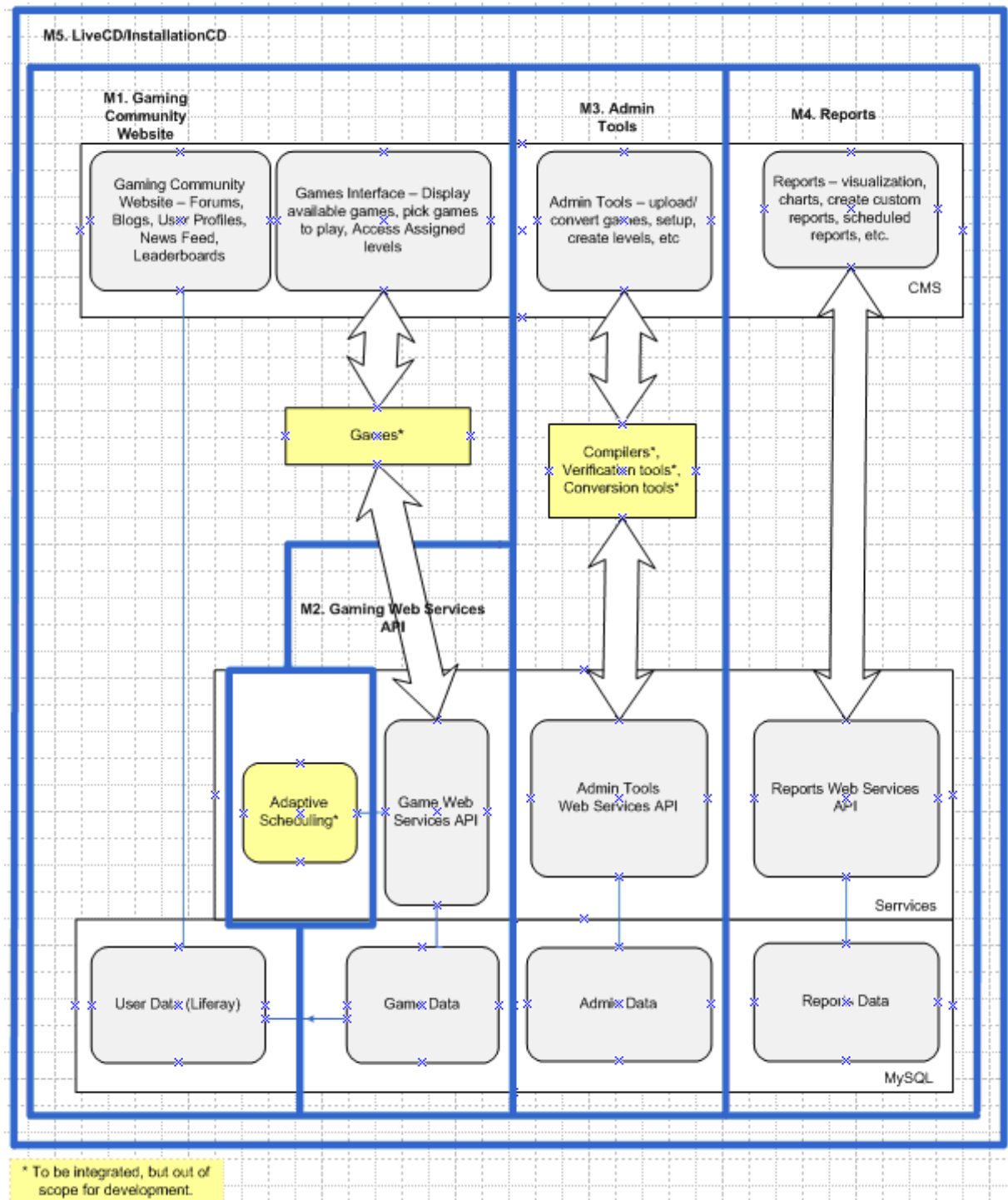
Capability C6 (monitoring and real-time reporting of Gaming Website activity) is achieved by a monitoring screen with application, game, and web-site usage “live” statistics displayed in real-time. All the reports on game verification results will be also available in real-time. There will be useful charts and detailed table with dynamically updated data on the verification progress on each software product, as well as detailed and summary statistics on user activity.

Capability C7 (distributing a stand-alone instance of the Gaming Website) is achieved by the Site Administrator. He/she can prepare distributions with the operating system, Gaming Website, tools, games, user manuals and troubleshooting guides. To the extent possible, this will be performed using LiveDVD technology for the Linux platform. The fully configured bootable DVD disk image will be generated using standard build tools, and it will contain a full installation of needed software to run and install the Gaming Website. A Site Administrator can boot from the LiveDVD and run the installation of the Gaming Website from that DVD.

Capability C8 (protecting privacy and users’ personal data) is achieved by minimizing stored users’ personal data – if possible, only e-mail address, username (handle), password, and birthday is stored. Additionally, only a numeric ID of the user and any personalization features (e.g., username, graphics, colors, etc.) is provided to the games, so the game will not have any unnecessary personally identifying information about the user. More protection is also achieved by using secure Internet connections (HTTPS) to access the Gaming Website, and by auditing of all the user actions in the application.

New System Features (modules M1 – M5)

The workflow overviews are grouped according to Modules M1 – M5 as shown in the following block diagram.



Module M1 (Gaming Community Website):

This module defines the most common functionality of the Gaming Website, the basic actions of the users and their communications. Read-only access to the application and several basic features will be available to all the users of the Gaming Websites, even if they are not-registered and Anonymous.

The Anonymous User can view the initial homepage (landing page) of the application. There will be a navigation menu, hyperlinks to top parts of the application and service information, some news, current games, etc. data on the homepage. The Anonymous user receives read-only access to forum posts, blog articles, and wiki pages that are publicly shared for all the community users. He/she can view information about currently performed short contest (like single game competitions between multiple users), and about long term on-going tournaments (like multiple game competitions between multiple users). Historical data of the contests and tournaments will be available too. The user can perform basic searching (like by keywords) for the needed content.

The Anonymous user most likely needs to get general information about the application abilities and usage, so he/she can freely get help, FAQ, Terms & Conditions, game play rules, and contact information for the application owners. Any web-page of the application can be freely printed by the user.

The Anonymous user can play some games, but he/she has to register to the application to have performance information saved for future play, to play more games and get access to more game levels on the Gaming Website. The registration process will be very simple, and the application will minimize usage of the personal data from the user (if possible just e-mail address, username and password, date of birth (or just handle and password from some social network) will be enough for registration). An e-mail notification will be sent to the user upon successful registration.

Each user will be initially registered just as a Registered User and a Game Player, but more user roles can be assigned to him/her by the Site Administrator.

The Registered User can login to the application to get more functionality from the Gaming Website. It will be possible to recover the user password if it was forgotten. The user can logout from the application at any time if needed.

The application will maintain the profile page for each registered user. The profile can be viewed by other users too, if the profile owner allowed sharing of his/her profile. The user can freely edit his/her profile, add more personal data (if desired), control sharing of his/her profile and own content to other users, and unregister from the application if desired (for example if the application isn't interesting for that user anymore).

The registered user will get his/her own dashboard, with customizable data such as contest results, news, announcements, messages, etc., and hyperlinks to the most popular features of the application.

The Registered User can perform various conversations, discussions and information exchange with other users of the application. He/she can post to own blogs and comment on other blogs, write data on the own personal wall and on the walls of the other users, post new threads and reply to existing threads on forums, produce wiki pages, perform textual chat with other users, post/like/re-publish data on the standard social networks (like Facebook, Twitter).

The application will provide interesting showcases (like visually appealing presentations) on the games and registered users can freely view that data before start playing the related games.

The user achievements (for the played games) will be shown for him/herself and for the other user. The new users will be especially encouraged by leader boards, scores, awards, badges and even physical prizes or gifts achieved by existing users. The Registered User may choose to be automatically notified (by e-mail) upon his/her new

achievements, new posts on watched forums, comments for his/her blogs, personal pages and wiki pages.

The Gaming Website is hosted on a web-server that is freely available to the user from the Internet. All the games will be published on the Gaming Website, so the user can find their pages, load them and play.

The System will provide needed communications between the public Gaming Website (installed on the Internet) and private instance of the Gaming Website (installed from LiveCD).

The System will integrate with mobile devices and allow access to the Gaming Website from common smartphones and tablets.

Integration of the Gaming Website with popular social networks will be supported by the system. Facebook and Twitter will be supported, so the user can easily “Like” the Gaming Website, or tweet some news from the Gaming Website, re-post some information about software verification games and his/her achievements.

The System will send e-mail notifications to the application users (like on new achievements, new/changed content, some events, etc.).

There will be a special General Web Services API providing programmatic access to the general functionality of the gaming web site and public community (like achievements, conversations, application general information, etc.). Some authentication will be required to access that API.

The System will automatically track all the statistics and analytical data on the application usage, users and games during work of the Gaming Website. The data is tracked to the database and can be used later when creating reports.

The System will automatically log all the errors, exceptions, warnings and debug information during the application execution. All the user actions, creations, modifications and deletions of the data will be audited by the system.

Module M2 (Gaming):

This module defines most gaming functionality on the Gaming Website. It allows the Game Player to choose and start the needed game, externally play it and, therefore, to perform actual indirect verification of the software source code. External games can interact with the application. This module is mostly about Game API and almost all functionality of the System for module M2 (shown below) is exposed through the Game API.

The Game Player can select which game to play from the set of the currently available software verification games. The Game Player will run the chosen game, it will be loaded and executed on his/her PC or the mobile device. The game is played according to the current game level given to the user. Playing the game is fully out of scope for this conceptualization – games will be provided externally.

The Game Player may attend special contests – like short competitions on a single game with multiple other game players. Those contests can have some predefined prizes/gifts or virtual awards/badges. It will be possible to participate in long tournaments (usually involved many game levels or even multiple games) and team competitions. The user is scored after the game completion and can be awarded by some achievements if he/she performed well and got high scores.

The System of the module M2 will integrate all the games to the Gaming Website via the Game API, so the user can load it and play either on his/her PC or on the mobile device. The system will perform needed integration between the Gaming Website and the native game played on the mobile device. All the functionality of the system of the module M2 (except logging, auditing, tracking analytics, and localizing GUI) is exposed through the Game API. The API will be secure and RESTful.

The system integrates special abilities of smart scheduling of game levels for playing by users with enough skills. Adaptive scheduling itself is out of scope for this conceptualization, but the infrastructure for integration will be provided.

The System can automatically calculate skills of each user based on their game play. That information will be used to determine which game levels can be allowed for playing by the related user. The system can adaptively assign more advanced levels to the Game Player as that user successfully finishes more and more game levels and gets high scores.

The system can automatically schedule games for a set of users by Adaptive Scheduling. The Adaptive Scheduling sub-system will determine users' skills, their performance and experience to decide which games and game levels are appropriate for the related users. It will be a smart algorithm allowing to maximize verification quality and speed of the verification process by selecting users appropriate to game and game level complexity. And the system will greatly scale playing a single game or game level through many Game Players. It means multiple Game Players will be able to play the same game in parallel to ensure that at least some of them will successfully finish and produce quite appropriate results.

The System will allow games to fully persist all the Gaming Website data to the database through the Game API. And it will support the needed input/output persisting actions on the data through the Game API. The System will load chosen games and provide them for playing by the users. The games can be played on user's PC or on the mobile device. All the results are automatically provided to the Gaming Website through the API and persisted during the game play.

The System will automatically determine user scores of the played games, calculate and apply user achievements (awards, badges, prizes/gifts).

The System will automatically have access to the results from played games and user solutions by receiving that data from games through the API. That data will be stored to the database and used later for generating verification reports. Any user achievements will be also persisted.

After a Game automatically determines when the game or it's level is completed (i.e. enough results and user solutions are collected), the game or game level completion status will be persisted to the database through the Game API.

There will be a special Game Web Services API providing programmatic access to the gaming functionality. It will allow game developers to easily integrate games with the Gaming Website. The API will include games and users' data transmission between games and the Gaming Website. Game levels and results will be transmitted as well through that API. Authentication will be required to access that API.

The Game API will also provide interfaces for verification tools to be able to extract game results data and user solutions, and store their output (verification output) back into the

application's storage. Game Developers (external) do not need to know internal data model of the Gaming Website, but they will simply follow the exposed Web Services API.

The System will track all the statistics and analytical information, log issues, and audit user actions – like described above for the module M1.

Module M3 (Admin tools):

This module is provides Administrator-related functionality. The Site Administrator can freely post news to the Gaming Website web-site and manage any of the content (like by creating, editing, removing any content entries). He/she can moderate new content posted by other users – such as forum posts, blog articles and comments, wiki pages and comments, posts on personal walls. Any inappropriate content can be edited or removed by the Site Administrator, and the related users can be managed (e.g., blocked or removed from the application).

The Site Administrator can manage user accounts. He/she can add new users, modify data of the existing users, disable/enable or even remove existing users from the application. He/she can modify access rights of users and assign them more user roles (or remove some roles from users). The Administrator can send special announcements to the chosen users by e-mail.

The Site Administrator can configure the application using GUI pages, so options such as e-mail server, paths to FTP servers, logging levels, etc. can be easily set up.

The Game Administrator can provide external special tools for converting software product's source code to the related game levels. The verification tools for conversion from achieved game results and user solutions for formal software verification output can be also externally provided by the Game Administrator to the Gaming Website. Some compilers or other tools needed for preparing games can be uploaded by the Game Administrator too.

Games are externally prepared and provided by the Game Administrator to the Gaming Website. The Game Administrator can configure the games by choosing setup options for the games (it is performed externally, like on the configuration pages specific to each game).

The Game Administrator can upload the source code of the software product to be verified to the Gaming Website. The Game Administrator can choose needed tools for the software product from the list of all the available tools. Some tools can be specific to programming languages (like Java, C), other can relate to special games, others can produce different formal output, etc.

The Game Administrator can choose the needed game level generators from the available tools and use them for generating the game levels from the source code of the software product to be verified. He/she can assign game levels to the related chosen Game Players – so it will be possible to manually define which game levels will be played by which users.

After completion of the game (or some levels), the Game Administrator can run a tool or tools to map collected user results and solutions to the formal verification output for the verified/evaluation software product. The tool will be developed externally and generate all the needed data for the software verification. While it also will be possible to get verification output reports automatically, Game Administrators can also obtain this information manually.

The Software Owner can provide his/her software products for verification and evaluation on Gaming Website. Source code of the software product can be simply uploaded by the Software Owner to the application. That data can be provided to the application in the offline mode too (externally).

The Software Owner can convert his/her software source code to the game levels by using external conversion tools. This might be needed so as not to expose source code of the software under test even to Game Administrators.

The Software Owner can download the formal output of the verification results on his/her software after all games are finished on his/her product. Or it will be possible to get formal output for parts of the software product already verified by the related game(s) and game levels. The Software Owner can download the formal output as documents from the application, or through the FTP server, or manually in the offline mode (externally).

The System of the module M3 will fully integrate with externally provided game tools – compilers, game level conversion tools, and verification tools.

There will be an Admin Tools Web Services API providing programmatic access to all the administration functionality. The API will allow for integration of new tools to the Gaming Website, and to manage the content, conversation, users, game tools, games, and software programs. Authentication will be required to access and use the API.

The System of this module will track statistics and analytical information, log issues, and audit user actions as described above for module M1.

Module M4 (Reports):

This module provides reporting functionality. Reports are produced for a special Reporter user role and to both administration user roles (i.e., Site Administrator, Game Administrator) – all of those users are referred to as Reporters in this module.

The Reporter can obtain various reports from the Gaming Website. The Software Verification Report is a finalized report on the verification of the software source code. It will provide both summary and detailed information about the results of the software product evaluation.

Reports on the usage of the application by various users, most popular pages and content will be provided. User statistics, preferences, and achievements reports will be provided. A report on game results, scores, optimal solutions, and other game statistics will be available.

It will be possible to customize reports by choosing desired parts of the report, e.g., columns, data sorting, filtering, etc. The user can choose the desired schedule for generating desired reports, so the application will automatically generate selected reports according to a schedule and persist them in the system.

There will be real-time reports on all the data of the application. Statistics, analytics, and verification output results will be dynamically displayed so the user can view “live” charts and tables of details and summary of the Gaming Website usage.

Reports will be available in the HTML, XML, CSV, and JSON formats. The application will display them and allow export in any of those formats (plus PDF and TXT) to a downloadable file. Any displayed report can be freely printed by a Reporter.

The System of module M4 can transform collected game results and user solutions to the formal verification output with the help of externally provided tool(s). Then, the formal verification output can be stored to the database.

The system will automatically send chosen scheduled reports to FTP servers so that they can be accessed later by other users (like Software Owners).

There will be a special Reporting Web Services API providing programmatic access to all the reporting functionality. It will allow to programmatic access to all the reports from the Gaming Website. Authentication will be required to access that API.

The system of this module will track all the statistics and analytical information, log issues, and audit all the user actions – like described above for the module M1.

Module M5 (Build and Installation DVD):

This module provides functionality for preparing, generating and installing private, stand-alone instances of the Gaming Website. The private instance will have all the functionality of the Gaming Website and can be locally used or installed from the LiveDVD.

The private instance can be installed (or simply run) from the specially prepared DVD (or CD) on any network by the Site Administrator. The operating system and private Gaming Website will be started from LiveDVD and the user can use the prepared private instance like a local (or offline) Gaming Website for performing private competitions within the customer's site.

All the actions of the Gaming Website (like described in modules M1 – M4) can be performed on the private instance. Data (like software products, game levels, formal verification results, etc.) can be interchanged between the public Gaming Website and the private Gaming Website according to the needs of private Gaming Website instance owners, and with appropriate permissions set on the public web site. The data can be transferred online (through some API), or in offline mode (using storage media).

The LiveDVD can be prepared by the Site Administrator through the GUI of the Gaming Website. The Site Administrator will prepare distribution of the needed OS (most likely a version of Linux with a web server) and select the desired distribution.

The Site Administrator can include tool distributions and games distributions on the LiveDVD. The user guide and troubleshooting guides can be provided on the LiveDVD too.

The application will have access to the build system for the Gaming Website, so that the Site Administrator can freely choose needed distributions and documents for generating the LiveDVD. The LiveDVD will be a bootable CD or DVD disk, which can be used for full installation of the Gaming Website of the customer's site. The LiveDVD can be also used to boot and immediately run the Gaming Website instance right from that CD or DVD disk, if the system has sufficient memory.

The System of module M5 can interchange data of the Gaming Website private instance with the public Gaming Website (if needed). There can be backend scripts for building the needed LiveDVD into some image file for further downloading by the Site Administrator.

The System of this module will track all the statistics and analytical information, log issues, and audit all the user actions – as described above for the module M1.

3.2 Context Diagram

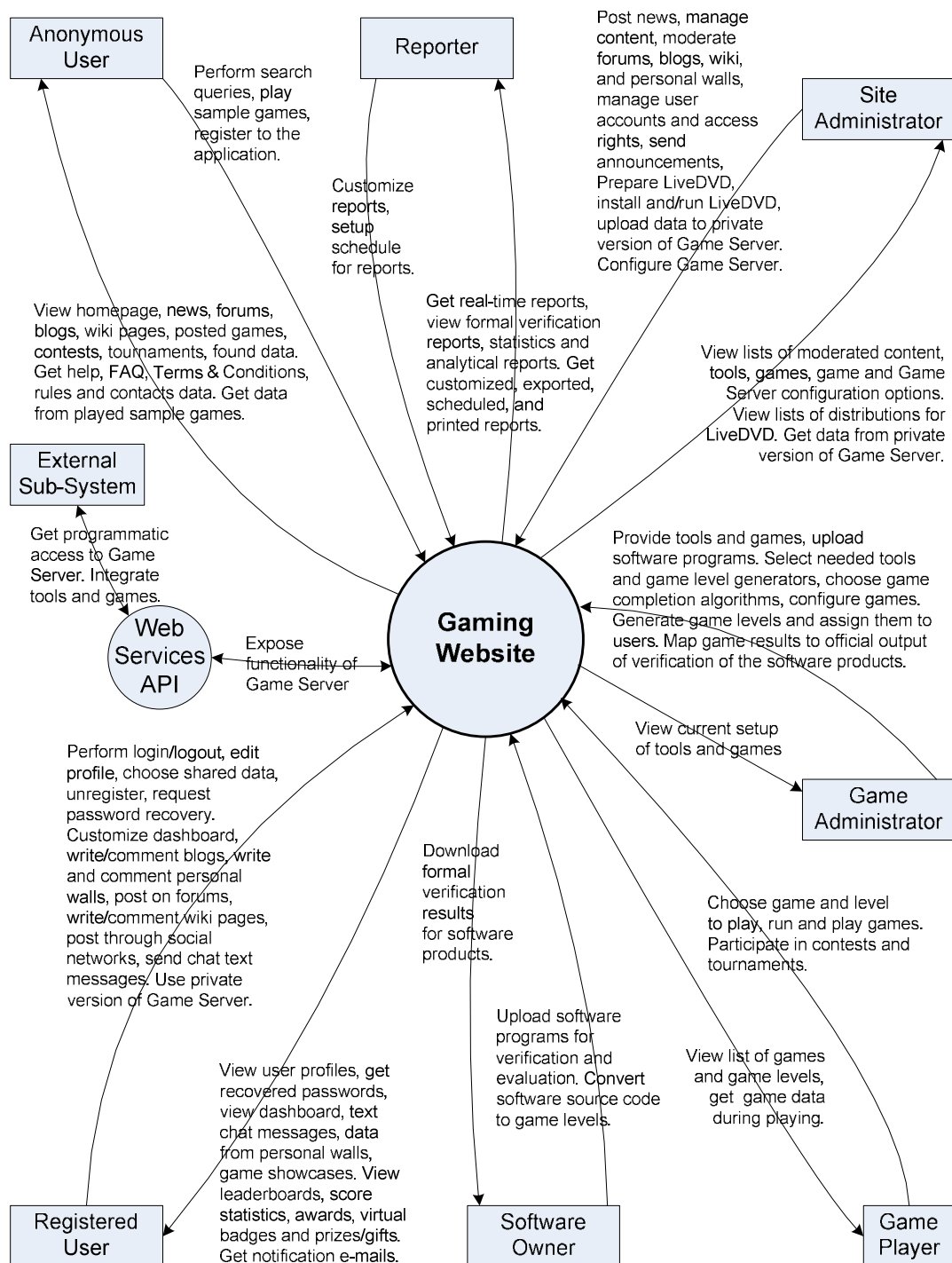
The Context Diagram illustrates the modules, business processes, etc that feed or interact with this proposed application.

The application is separated into the following modules:

- M1 – Gaming Community Website,
- M2 – Gaming,
- M3 – Admin tools,
- M4 – Reports,
- M5 – Build and Installation DVD.

Many dependent systems relate to multiple modules, so it isn't possible to show modules on the context diagram. Please refer to the use cases (chapter 3.4), which are divided by modules.

Context Diagram: Proposed Process (Users). CSFV Community Gaming Website 1.0



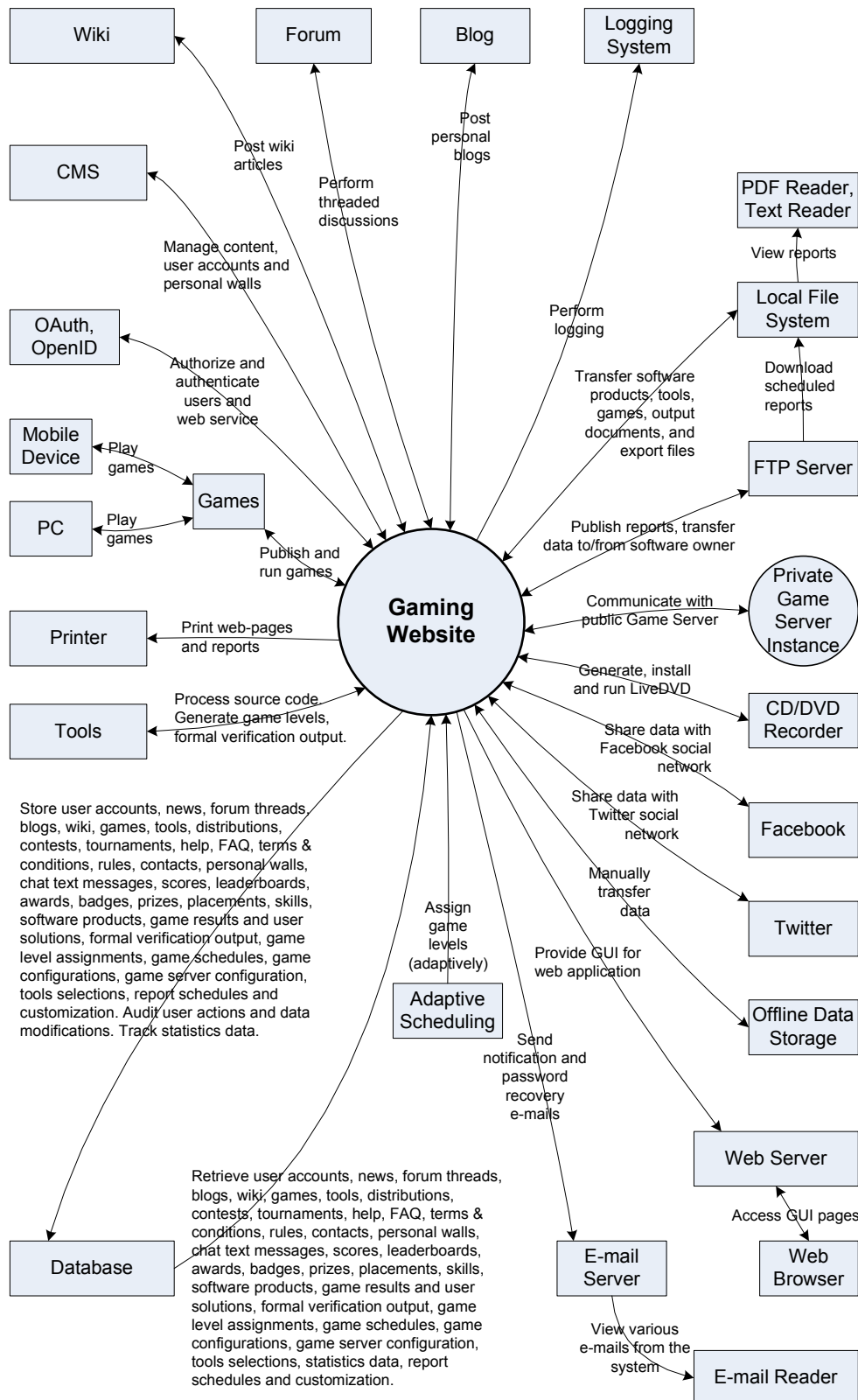
Please note

- The Registered User can perform all actions of the Anonymous User.
- The Game Player, Reporter, Software Owner can perform all actions of the Registered User. Game and Site Administrators can perform all actions of the Registered User.

Please note

- The actions on the diagram are described from the users point of view.

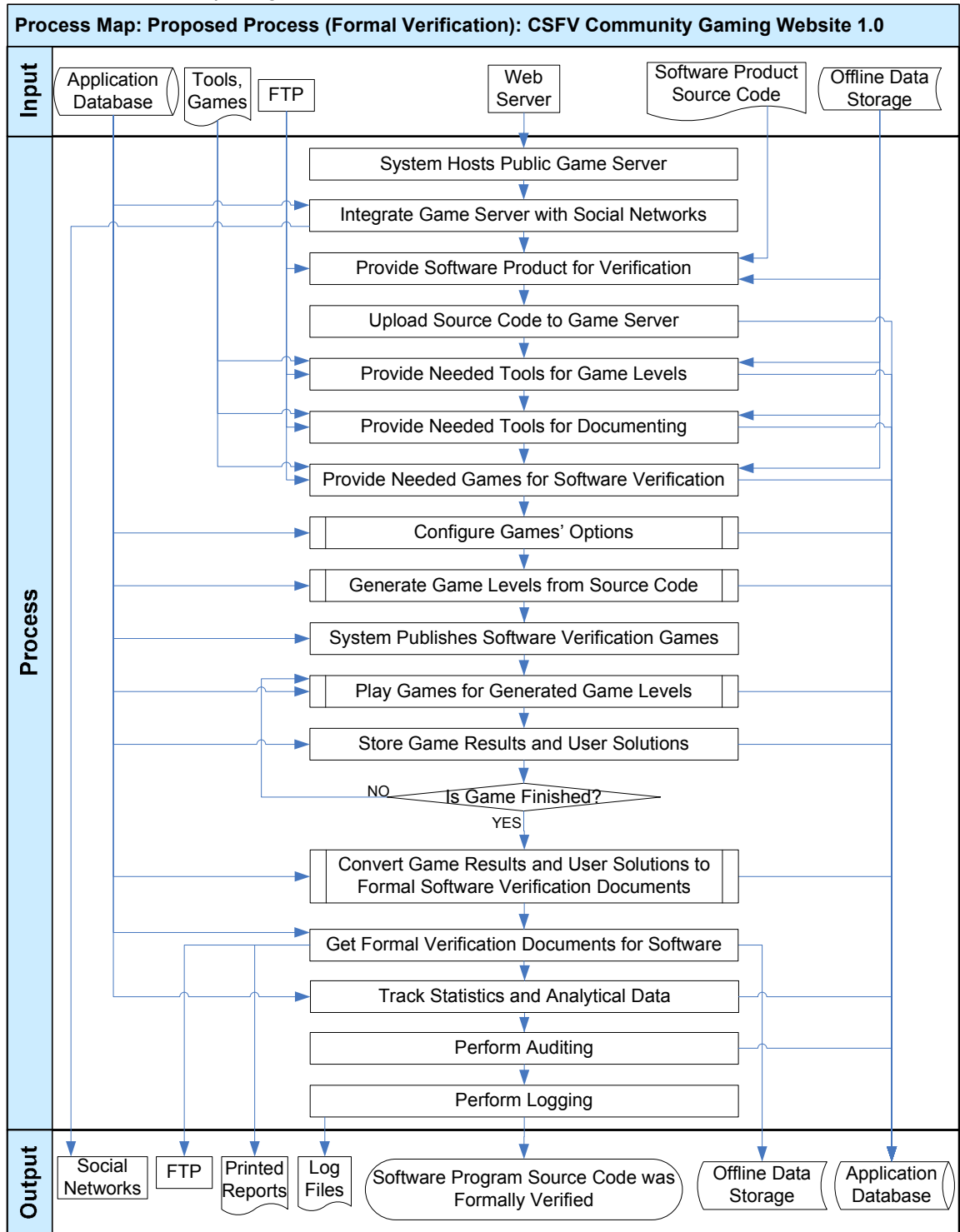
Context Diagram: Proposed Process (Systems). CSFV Community Gaming Website 1.0

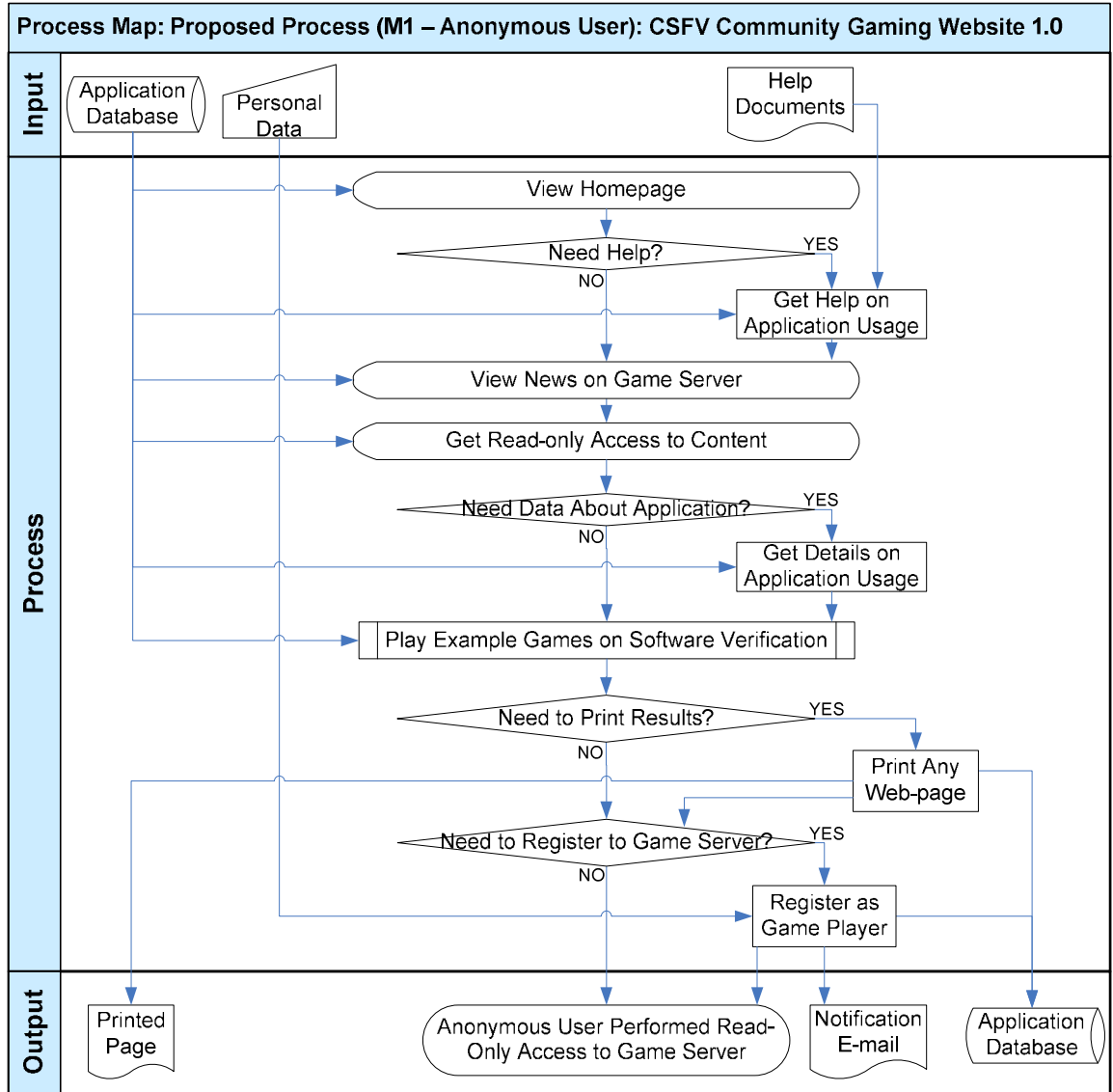


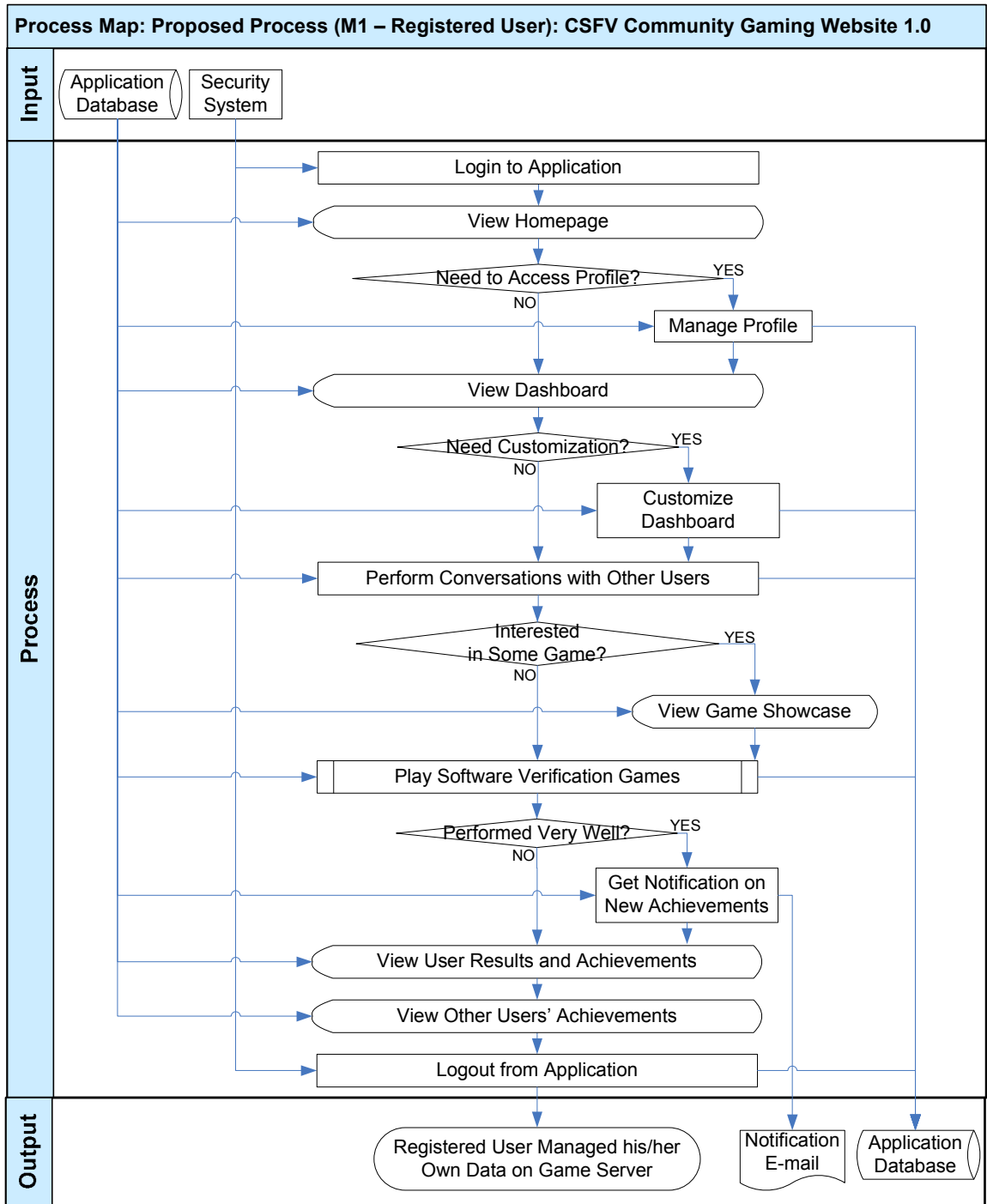
3.3 High Level Workflow

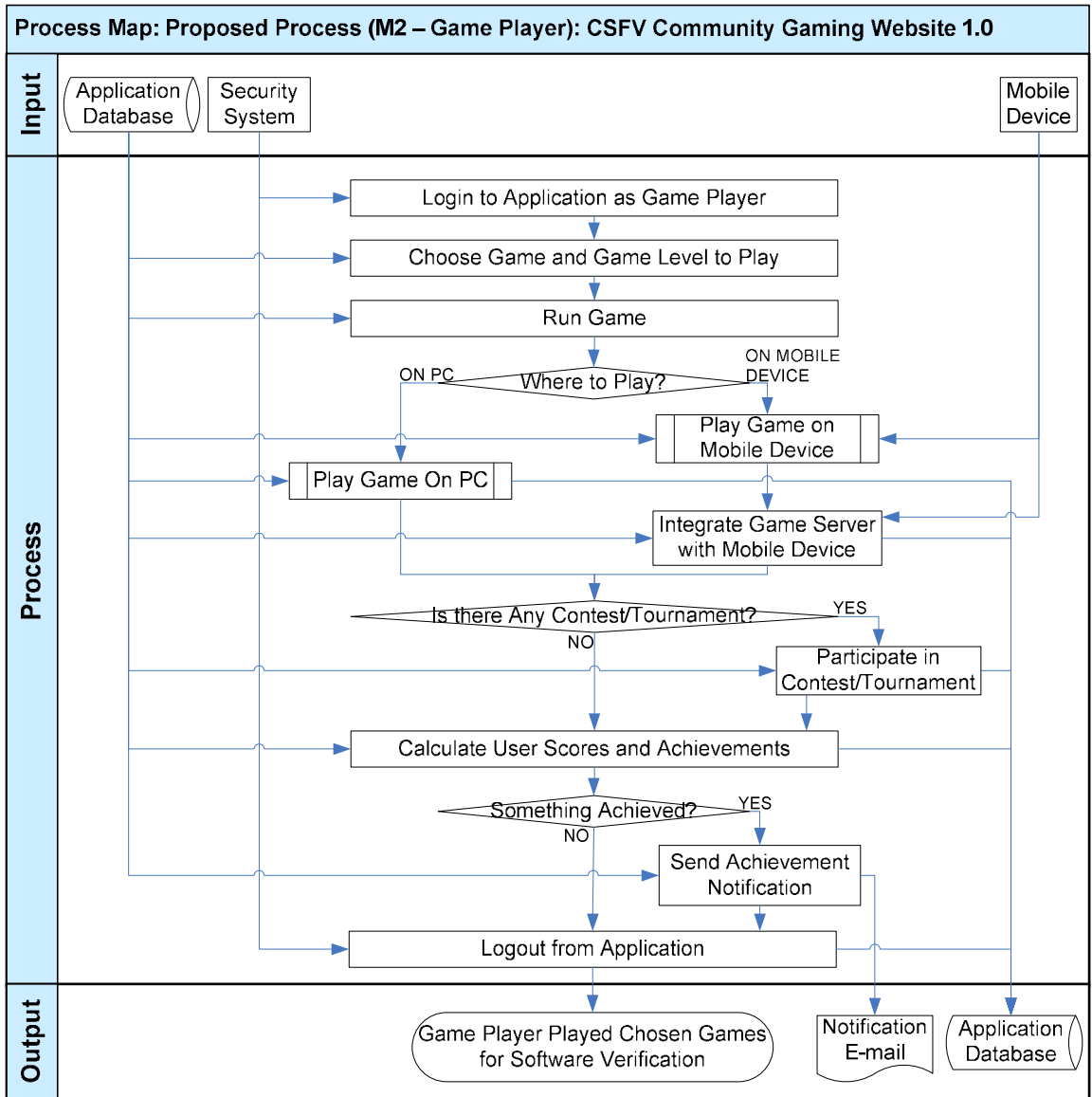
The workflow required to complete the primary objectives of the proposed application is described below. The workflow is business-centered, and includes “decision forks” for decisions the business user, or application, must make to achieve the objective. The workflow omits application faults or exceptions. Individual tasks in the workflow are described.

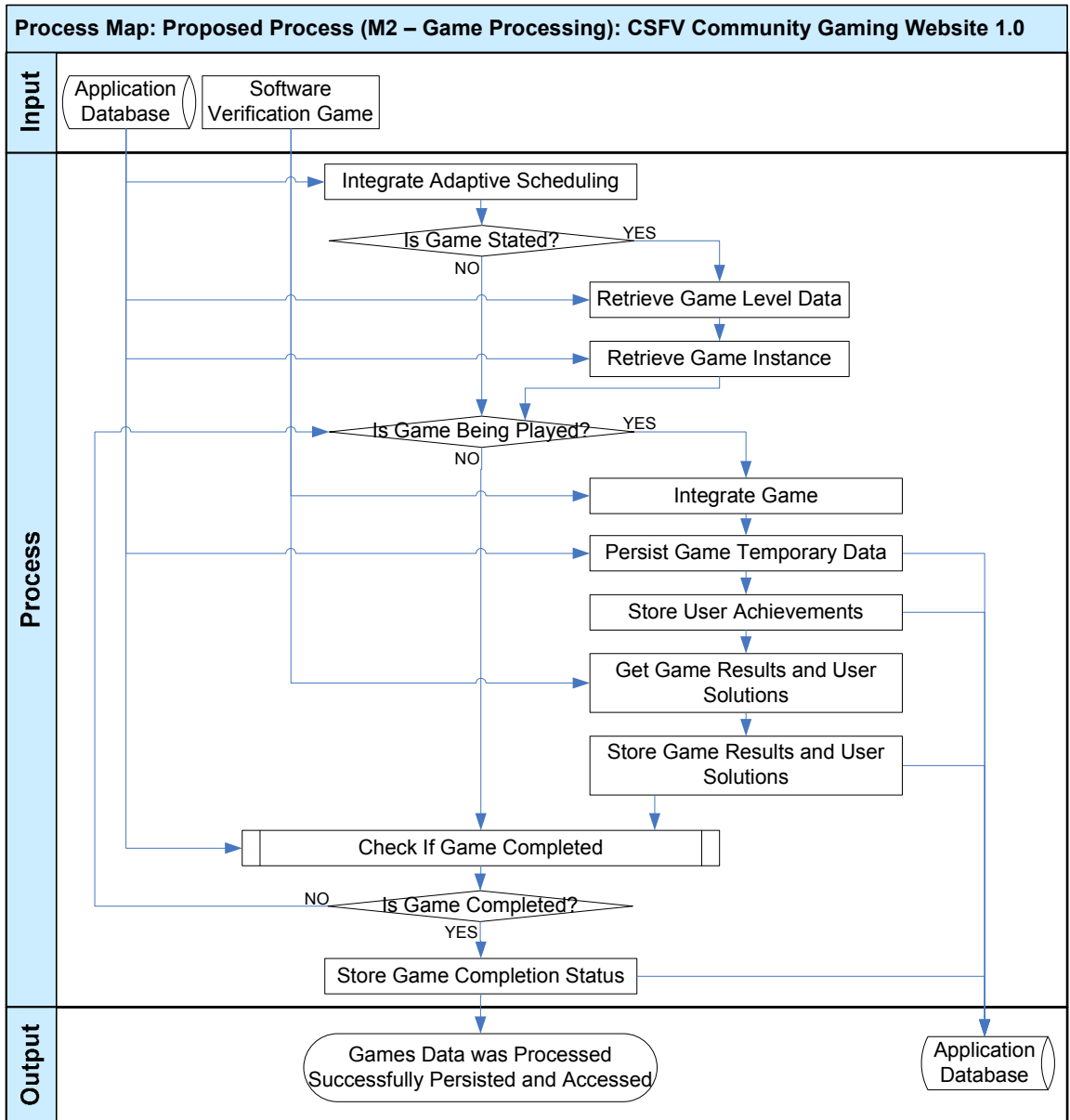
3.3.1 Workflow/Process Map Diagram

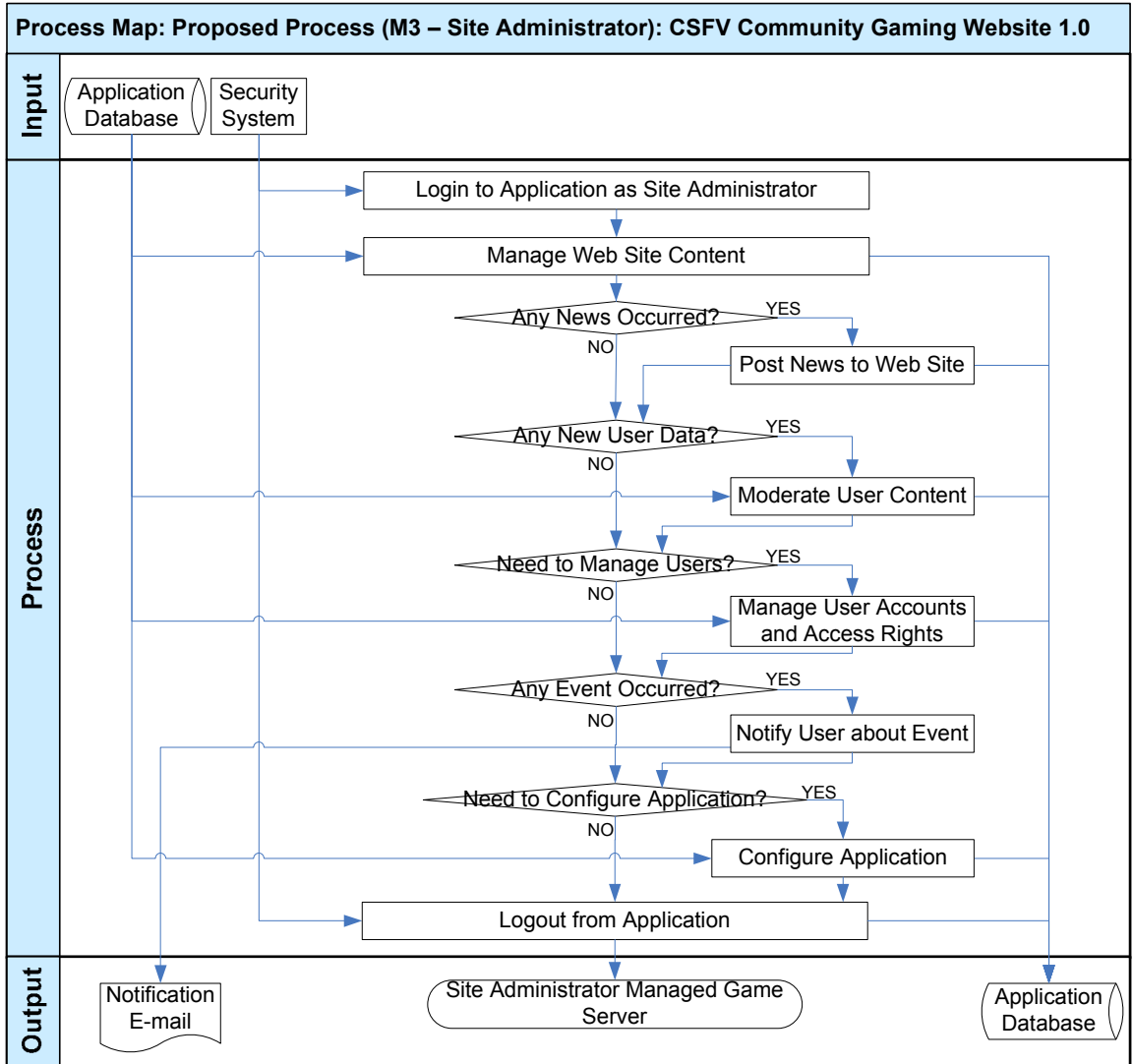


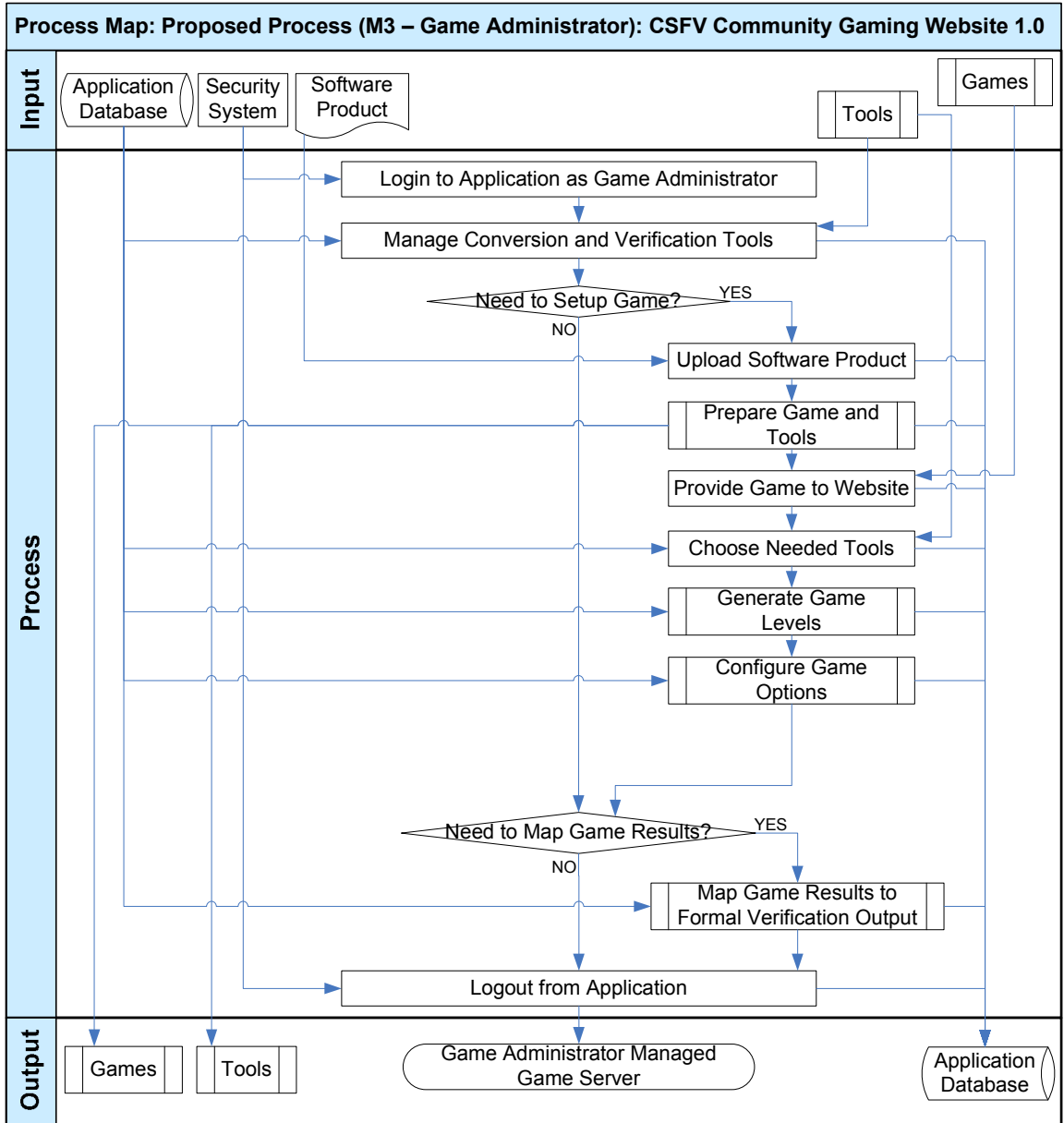


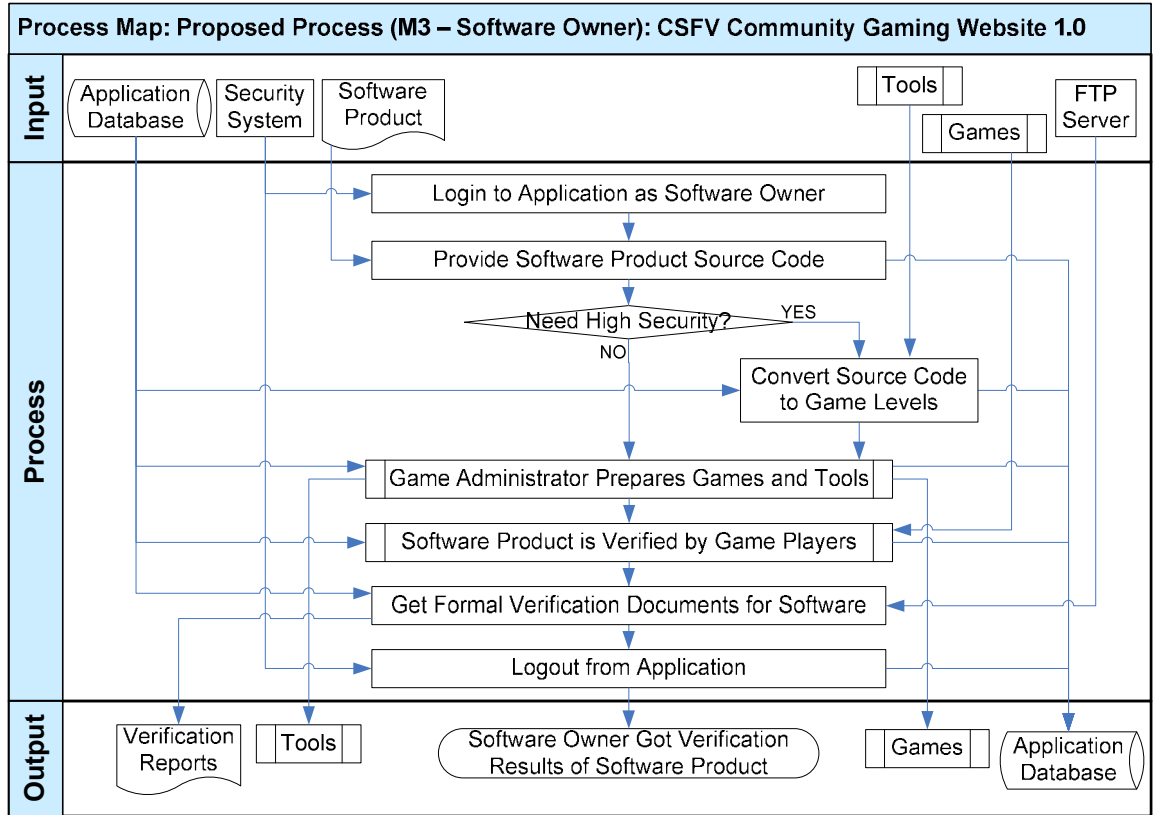


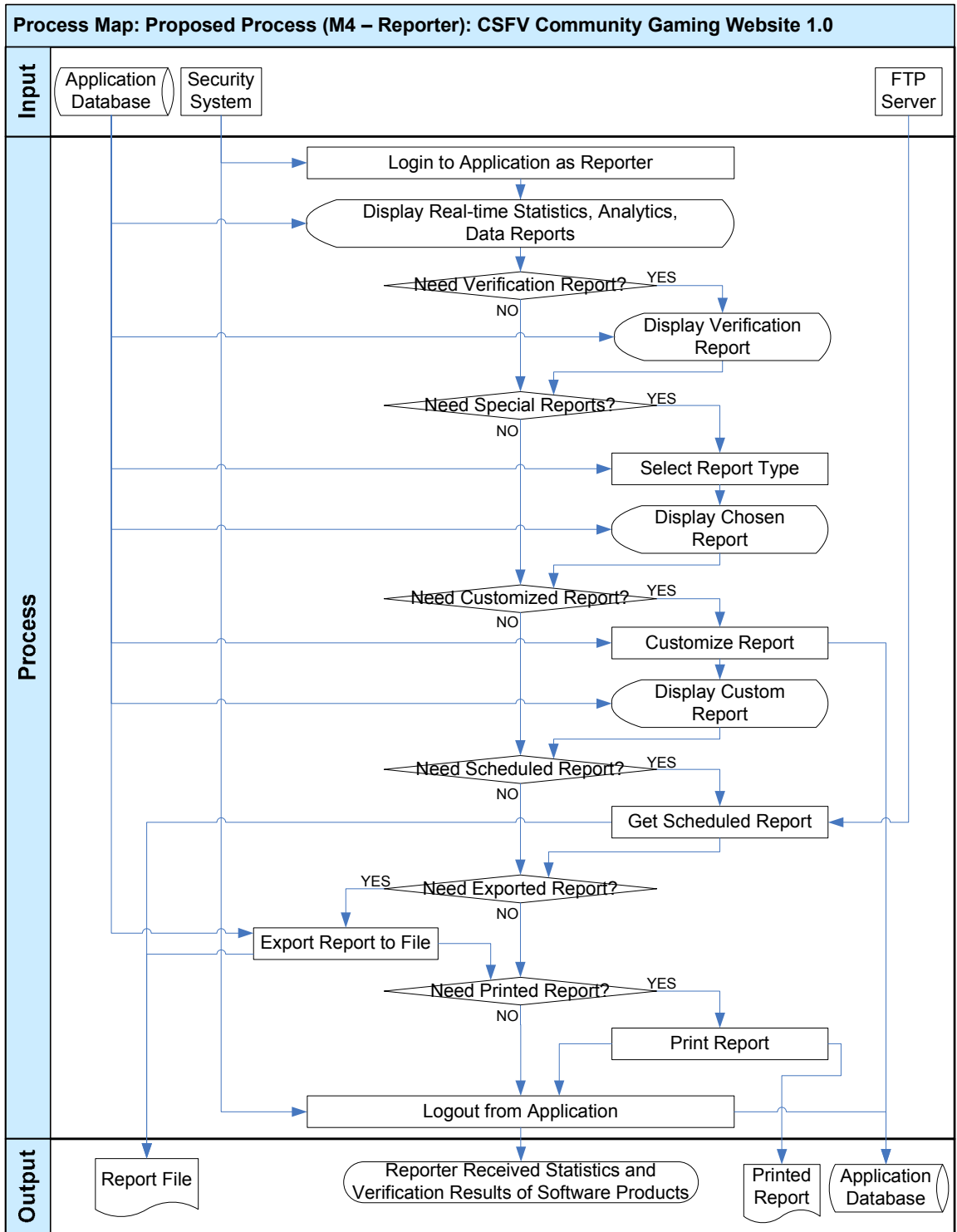


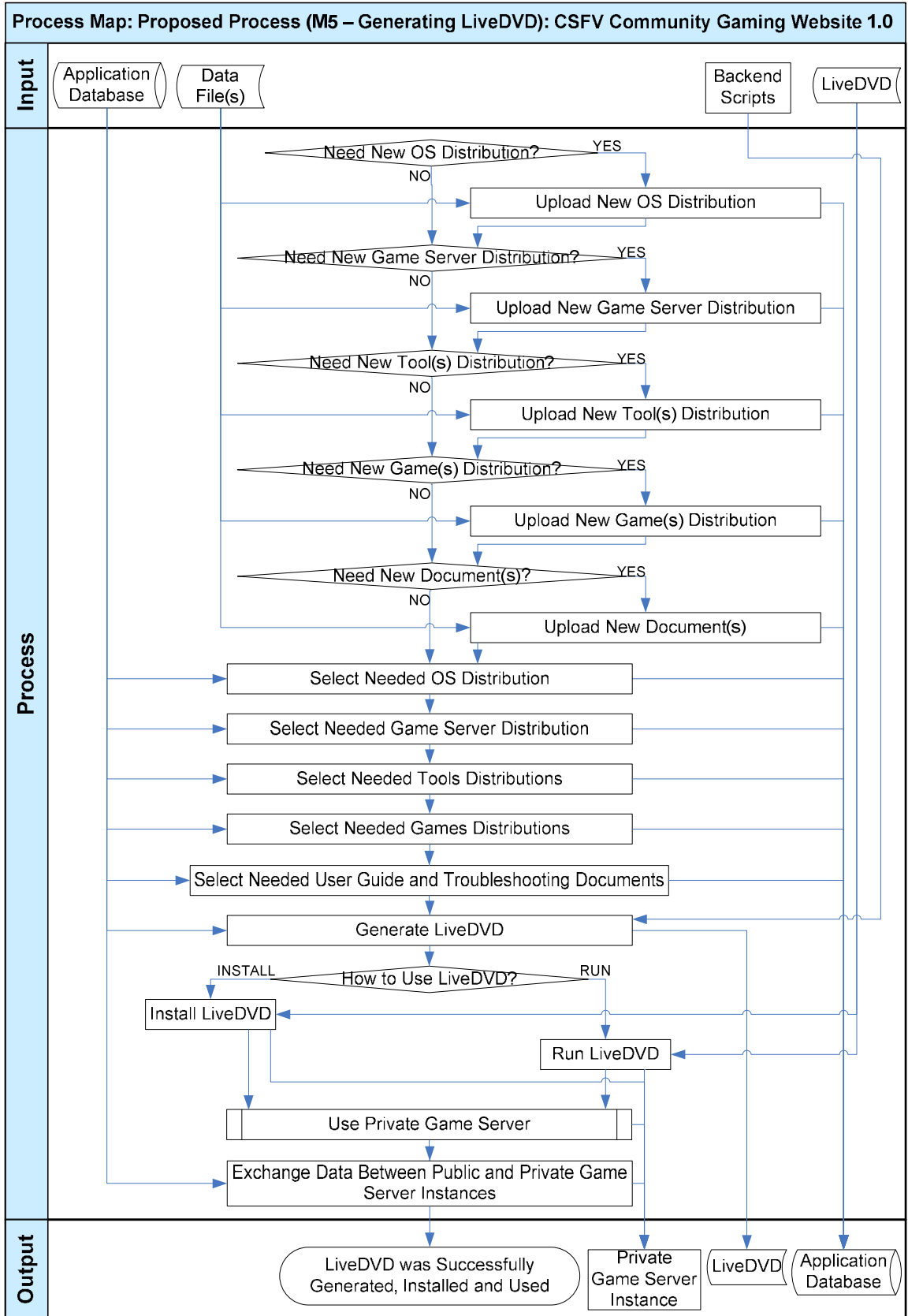


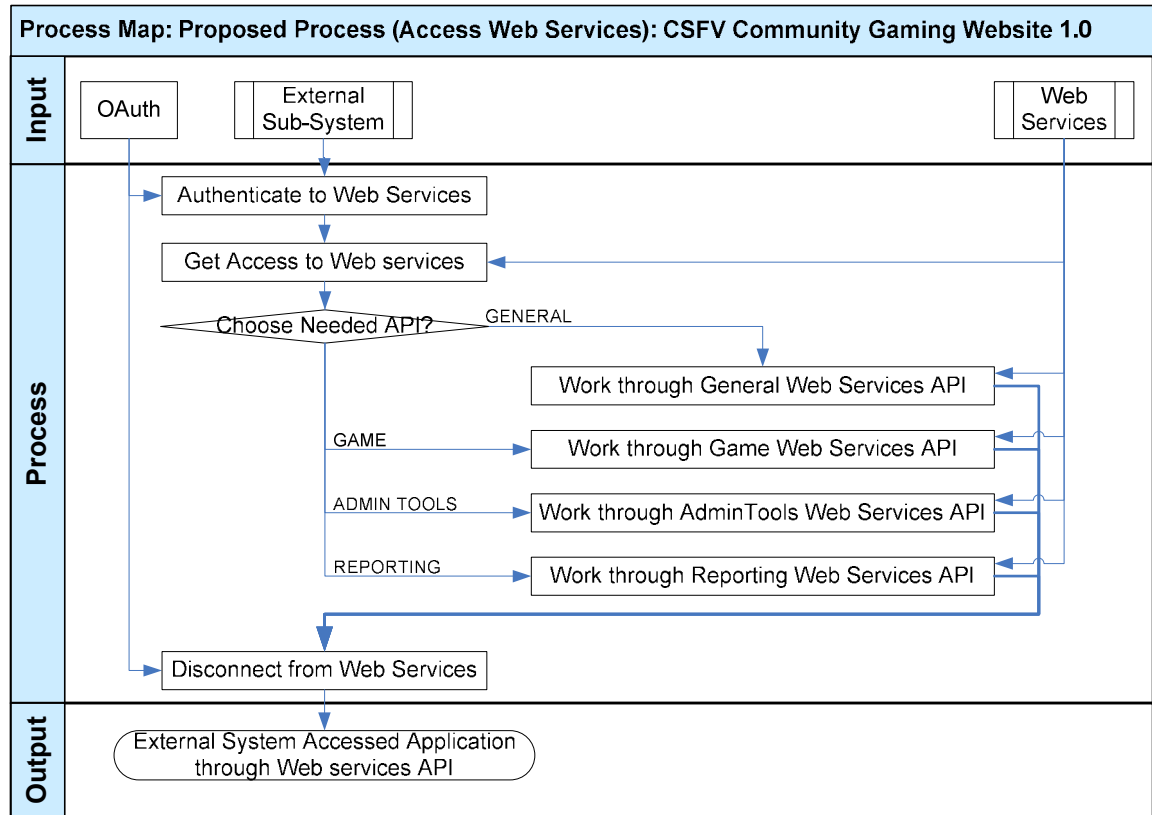












3.3.2 Workflow Description

Formal Verification:

Please note, this is the workflow of performing formal verification and evaluation of the source code for software products. This workflow is performed on the top of the Gaming Website web-site and relates to multiple modules (M1 - M4).

1. The System will host the Gaming Website on a dedicated web-server, so it will be freely available to users from the Internet.
Possible opportunity: to provide public access to software verification process.
2. The system will perform needed integration between the Gaming Website and popular social networks.
Possible opportunity: to share Gaming Website data on social networks for increasing user interest and for marketing the site.
 - a. Popular social networks will be supported by the Gaming Website. Specifically, Facebook and Twitter will be supported, so the user can easily "Like" the Gaming Website, or tweet some news from the Gaming Website, re-post some information about software verification games and his/her achievements.
3. The software product to be verified and evaluated is provided by the Software Owner.
Possible opportunity: to support multiple useful ways for software product source code delivery for verification on the Gaming Website.
Possible problem: there can be a problem that software product to be evaluated is too complicated or too special to be properly verified by the Gaming Website. Or the source code can have mistakes, or it can be incomplete, corrupted, etc. The application has to inform Software Owner about any problem found with respect to the provided source code.
 - a. The Software Owner can deliver the source code of the software product to the Gaming Website through an API or with a FTP server.

- b. The Software Owner can provide source code of the software product directly to the Gaming Website through GUI pages. So, he/she will directly upload the source code to the Gaming Website.
 - c. The Software Owner can also manually provide source code of the software product to the Game Administrator on data storage media. It means providing data in the offline mode.
- 4. Source code of the software product is uploaded to the Gaming Website – either in online mode (through API, or GUI page of the application), or in offline mode.
Possible problem: there can be a problem that source code is too large, so some progress indicator is needed during uploading to inform the user.
Possible bottleneck: there can be a performance bottleneck if too many large software products are being simultaneously uploaded to the Gaming Website. In this case, uploading of new software products will be queued until previously started uploading processed is finished.
 - a. In the second case, the Game Administrator will upload the manually delivered source code to the Gaming Website. It will be performed on some GUI page of the application.
- 5. The Game Administrator will provide the needed tools for processing uploaded source code to the Gaming Website. Please note, all tools are externally developed and are out of scope for this conceptualization.
Possible opportunity: to allow flexible and easily customizable use of tools for game level generating.
Possible problem: the uploaded tools can be incompatible or just partially compatible with the software product.
 - a. The Game Administrator can provide tools for game level generation to the Gaming Website.
 - b. The tools can be simply uploaded through the GUI page.
 - c. The Game Administrator will select which game level generator tools to use for the current software product under verification.
- 6. The Game Administrator will provide the needed tools for converting played game outputs to the formal verification documentation to the Gaming Website. Please note, all tools are externally developed and are out of scope for this conceptualization.
Possible opportunity: to allow flexible and easily customizable usage of game results processing tools.
Possible problem: the uploaded tools can be incompatible or just partially compatible with the software product.
 - a. The Game Administrator can provide tools for mapping game results and user solutions to the formal output verification documents.
 - b. The tools can be simply uploaded through the GUI page.
 - c. The Game Administrator will select which documenting tools to use for the current software product under verification.
- 7. The Game Administrator will provide the needed software verification games for the software product under verification to the Gaming Website. Please note, all games are externally developed and are out of scope for this conceptualization.
Possible opportunity: to allow flexible and easily customizable use of software verification games.
Possible problem: the uploaded games can be incompatible or just partially compatible with the software product.
 - a. The Game Administrator can provide interesting and visually appealing games for verification of the software product.
 - b. Games can be provided for usage on PC and for mobile devices as well.
 - c. The games can be simply uploaded through the GUI page.
- 8. The Game Administrator can freely configure options of the uploaded games.
Possible opportunity: to easily setup games and adjust their parameters in user friendly manner.

- a. Games will be configured on their GUI pages – externally developed together with games (they are out of scope for this conceptualization).
9. The Game Administrator (or Software Owner) will generate needed game levels for the games of the software product under verification.
Possible opportunity: to automatically prepare custom game levels according to the source code of the software product under verification.
Possible problem: there can be a problem that automatically generated game levels are too difficult/complicated, or boring, or non-manageable, so human Game Players will not be able to complete them.
Possible problem: there can be a problem that generating game levels takes too much time. Some progress indicator is needed to inform the user about ongoing process.
Possible bottleneck: there can be a performance bottleneck if too many large software products are being simultaneously processed for game levels generation. Some generating tools can be queued until previously started generating tools are finished.
 - a. The Game Administrator will select proper tools for generating needed counts of levels for each game for software product verification. Please note different tools can be used for generating different levels even for the same game.
 - b. The Adaptive Scheduling will attempt to provide appropriate game levels for users.
 - c. The Game Administrator will generate needed game levels for the source code by using the chosen tools – they are implemented externally and level generation is out of scope for this conceptualization.
10. The games will be published on the Gaming Website, so the user can load them and play on either on his/her PC or on the mobile device.
Possible opportunity: to provide web-pages for each game and allow easy locating of those games on the web-server.
 - a. The games will be developed externally (out of scope for this conceptualization). They will be uploaded and listed on Gaming Website.
11. Game Players will be able to play prepared game levels as needed.
Possible opportunity: to verify source code in the game-like environment.
Possible bottleneck: there can be a performance bottleneck if there are too many simultaneously playing users – their actions can be queued and some delays can occur during game screens (according to queued data transfers).
 - a. Games are implemented externally (by Game Developers – out of scope), so playing them is out of scope for this conceptualization.
 - b. The architecture should take into account the potential performance problems.
12. All the data achieved by the Game Players during playing the games are stored to the database and they will be accessible through the API.
Possible opportunity: to automatically acquire and persist the verification results.
 - a. All game results (i.e. user output in game play) are stored to the database.
 - b. All optimal solutions (found by users in game play) are stored to the database.
13. The application will allow Game Players to play games for software product under verification until they are finished (i.e. proceed to step 11 until completion the game).
14. The system will automatically convert game results and user solutions to formal verification output for the software product under verification, and the Game Administrator can customize that conversion.
Possible opportunity: to automatically generate formal verification output for the software products according to collected game data.
Possible problem: collected game results and user solutions may be incomplete or incorrect, so the generated documents will not fully cover verification and evaluation details of the software product and can contain mistakes.

Possible problem: there can be a problem that transformation of game data takes too much time. Some progress indicator is needed to inform the user about ongoing process.

Possible bottleneck: there can be a performance bottleneck if too many large software products are being simultaneously processed for formal verification output generation. Some generating tools can be queued until previously started generating tools are finished.

- a. That generating of formal output will be performed externally by the related tool in the automatic way.
 - b. The Game Administrator can use the chosen tool for custom mapping of game results and optimal found user solutions, so formal documents with software product verification results can be generated in the customized way.
 - c. The generated formal documents are stored to the database.
15. The Software Owner can get software verification documents from the Gaming Website.

Possible opportunity: to deliver formal verification documents for the software products to the related Software Owners.

- a. He/she can download the generated formal verification documents through an API (or from the FTP server).
 - b. He/she can download the generated formal verification documents on the related GUI page of the Gaming Website.
 - c. He/she can manually get the generated formal verification documents on the offline data storage (like by asking Game Administrators to manually deliver that data).
16. The System will automatically track statistics on application usage, users and games. This is shown at the end of the workflow for clarity, but it actually will be performed in ALL the other steps of this workflow. This step is present in all the following workflows too, but it will be omitted for clarity and to reduce other workflows' sizes.
- Possible opportunity:** to collect data about the application, so reporting and analysis of the application use can be performed on that data later.
- a. The data is stored in the database and can be used later when creating reports for Reporters and administrators.
17. The system will automatically log the errors, exceptions, warnings and debug information during the Gaming Website application execution. Logging will be performed to some sort of the logging system and the information can be stored to log files. This is shown at the end of the workflow just for clarity, but it is actually performed in ALL the other steps of this workflow. This step is present in all the following workflows too, but it will be omitted for clarity and reducing of other workflows' sizes.
18. The system will automatically audit user actions and data modifications. New data creation, removal or update of the existing data will be automatically audited in the related database tables. Audited data will be used later for the application maintenance, analysis and improvement. This is shown at the end of the workflow just for clarity, but it is actually performed in ALL the other steps of this workflow. This step is present in all the following workflows too, but it will be omitted for clarity and reducing of other workflows' sizes.

M1 – Anonymous User:

Please note, the anonymous users will have more limited access to the system than registered users.

1. The anonymous user will come to the homepage have access to general information and hyperlinks to application features. Please note, registered users will also get homepage that contains data relevant to his/her user role.
- Possible opportunity:** to effectively visualize the most needed application's features to the user according to his/her role, so he/she can quickly and easily access them.

- a. The most important data and most popular functionality will be directly available on the user's homepage.
 - b. There will be a navigation menu, hyperlinks to top parts of the application and service information, some news, current games, etc. data on the homepage.
2. If the user needs help on the application usage, then he/she can easily get it.
Possible opportunity: to clarify application usage for the user and allow him/her to quickly and fully understand the application interface and abilities.
 - a. Brief help on the application usage will be shown for the user.
 - b. Some visually attractive help (like getting started) is needed, so the user will be quickly and easily get involved in the usage of the application.
 - c. The user guide and troubleshooting guides can be also retrieved by the user.
3. The user will view news and events related to the Gaming Website.
Possible opportunity: to highlight new events and possibilities of the Gaming Website to the users.
 - a. The news will be briefly listed (like the header and the first paragraph).
 - b. The user can choose a desired news entry and view details.
4. The anonymous user will have read-only access to content that is publicly shared for all the community users.
Possible opportunity: to allow users to view and read most content on the Gaming Website.
 - a. The user can view forum threads and posts from the community users.
 - b. The user can view blog articles and comments from the community users.
 - c. The user can view wiki pages and comments from the community users.
 - d. The list of posted games will be available for the user too.
 - e. The user can view information about currently running short contests (like single game competitions between multiple users).
 - f. The user can view data about long term on-going tournaments (like multiple game competitions between multiple users) and team competitions.
 - g. Historical information of the contests and tournaments will be available too.
 - h. The user can perform basic searching (like by keywords) for the needed content.
5. If the user needs to get more information and service data about the application, then he/she can see it through standard topics as following:
Possible opportunity: to provide standard pages on the web-site usage.
 - a. The user can view FAQ page with information about most common questions and answers.
 - b. The user can view Terms & Conditions page with legal information on the application usage.
 - c. The user can view rules of Games Server and software verification approach.
 - d. The user can view contacts data of the Gaming Website and support team. Information about DARPA, the CSFV program, and the performers will be shown too.
6. The anonymous user can play games but there will be limitations on what information is saved.
Possible opportunity: to allow users to investigate software verification gaming without registration and to provide easy learning of Gaming Website.
 - a. Only a limited number of games or levels will be available without registration.
 - b. The games will be played externally – on the user's computer or on the mobile device.
7. If the user needs a hardcopy or any data, content, scores, games, etc. displayed on the screen, then he/she can print that data page on the printer.
Possible opportunity: to allow creating printed reports on various data of the application.
 - a. The user's web browser can print the web page content.

- b. It will be possible to print a chosen table on multiple pages, so full large reports can be easily printed.
- 8. The application will allow anonymous users to register to the Gaming Website.
Possible opportunity: to allow registration and access to more functionality and games on the Gaming Website.
Possible problem: there can be a problem that some user can try register multiple accounts, or automatic registrations from spammers can be performed.
 - a. Registration process will be very simple, and the application will minimize usage of the personal data from the user (just e-mail address, username and password, date of birth (or just handle and password from some social network) will be enough for registration).
 - b. An e-mail notification will be sent to the user to confirm the email address, and as a protection against registration spam.
 - c. Registration will provide the user access to more functionality of the Gaming Website.
 - d. The user will be registered just as a regular Registered User and a Game Player, but more user roles can be added later to his/her account by the Administrator.

M1 – Registered User:

- 1. The user will login to the Gaming Website application by providing his/her credentials.
 - a. Login can be done via OpenID or the secure login mechanism of the website.
 - b. The application will authenticate and authorize the user according to his/her user role (i.e. Game Player, Site Administrator, Game Administrator, etc.) and provide the related functionality.
 - c. Each user will get data access in the application according to his/her user role.
- 2. The application will display the homepage with the data and links to features, related to the user role of the current user.
Possible opportunity: to effectively visualize the most needed application's features to the user according to his/her role, so he/she can quickly and easily access them.
 - a. The most important data and most popular functionality will be directly available on the user's homepage.
 - b. There will be a navigation menu, hyperlinks to top parts of the application and service information, some news, current games, etc. data on the homepage.
- 3. The registered user will be able to quickly get access to his/her own account on the Gaming Website.
Possible opportunity: to allow full management of own account and viewing of other community users' account by the registered user.
 - a. The user can freely view his/her account and all the data entered by him/her to the account.
 - b. It will be possible to search for and view accounts of other users that shared their data on the Gaming Website. It will be possible to open the user account by pressing hyperlink on the username of the related user on the Gaming Website.
 - c. The user will be able to edit his/her account information. The user can choose which data from his/her profile to share with other community users, and which data records from his/her profile have to be hidden.
 - d. The user can unregister from the application if needed (like to terminate his/her account and to never more login to the application). The site
- 4. The registered user can open his/her customizable dashboard with the widgets, related to most recent data (like forum posts, chat messages, news, announcements, etc.) for that user.

5. If the user needs to modify layout and placements of widgets on his/her dashboard, then he/she can easily customize it.
Possible opportunity: to provide a useful page for monitoring latest events, notifications and messages from the Gaming Website.
 - a. The user can perform standard actions with the dashboard widgets – like moving, re-sizing, selecting page layout.
 - b. It will be possible to add or replace visible dashboard widgets from a list of available ones.
6. The Registered User can perform various social networking activities and participate in conversations, discussions and information exchange with other community members. Registered users can also invite non-members to register on the site. Registered users may also invite other registered members to become part of their network of friends.
Possible opportunity: to provide standard social networking features for the users, so community members can easily communicate each to other, spend more time on the website, and build a community of players.
Possible problem: there can be a problem that some user-posted content can be inappropriate or illegal.
 - a. The user can post articles on his/her blog and place comments to blogs of the community members.
 - b. The user can write data on his/her own personal wall and on the walls of other community members.
 - c. The user can create new forum threads and post, reply to posts from other community members.
 - d. The user can create articles on the wiki and comment articles of other community members.
 - e. The user can perform data exchange with common social networks (like Facebook, Twitter) – i.e. posting some game-related articles, “like” Gaming Website, re-publish data, etc.
 - f. The users will also be able to leverage their existing social networks to draw more users to the site by sending invitations to those contacts.
 - g. The user can engage in real-time group chat, or one-on-one messaging which will be archived as message threads.
7. If the registered user is interested in some game for software verification, then he/she can view the specific game’s main page.
Possible opportunity: to demonstrate and promote the game, so users of the Gaming Website will be able to learn more about the game.
 - a. The page will display the visually appealing brief presentation of the chosen game, its real-time leader board, rules, screenshots, etc.
 - b. A picture gallery, video gallery, tutorial section might also be part of this page.
8. The registered user can play software verification games, published on the Gaming Website. Please refer to the workflow “M2 – Game Player” below for more details.
Possible opportunity: to verify source code of software products in the game-like environment by registered user.
9. Users can earn awards and/or badges for overall and game specific achievements and/or performance.
Possible opportunity: to keep the user engaged, recognized, and motivated.
 - a. The Gaming Website will notify the user when rewards or badges are earned, and if applicable, display them on the user’s profile.
 - b. Badges earned can be exported to a backpack via the Mozilla Open Badges framework.
10. The registered user can view his/her gaming results/achievements for games, contests and tournaments.

Possible opportunity: to fully inform the user about his/her performance on the software verification games.

- a. The application will display the user scores, ranking and placements.
 - b. The application will display user awards (like top places, fast speed, the best optimal solution, etc.).
 - c. The application will display user badges – like some small pictures representing the user skills, experience, and exciting winnings.
 - d. The application will display any physical prizes or gifts earned by the user for paid games, contests or tournaments, and team competitions.
11. The registered user can view results/achievements for games, contests, and tournaments performed by other community members.

Possible opportunity: to motivate users by performance of other community members and to support competing between users.

- a. The user can view leader boards with summary skills of community members. There can be a single common leader board, and special leader boards per each contest, tournament and game.
 - b. The user can view scores, placements, awards, and badges for other community members.
 - c. Information about prizes/gifts earned by other community members will be only viewable if those community members agreed to share that personal data.
12. The user can log out from the application to ensure high security – this application provides quite sensitive data to him/her, so the user has to logout as soon as he/she no more needs to access the application

M2 – Game Player:

1. The Game Player will login to the Gaming Website by providing his/her credentials.
 - a. Login will be performed via secure login or OpenID.
 - b. The application will authenticate and authorize the user according to his/her user role (i.e. Game Player) and provide the related functionality.
 - c. Each user will get data access in the application according to his/her user role.
2. The Game Player can select which game and level to play from initially the set of the currently available software verification games.
 - a. Initial game levels can be chosen by the user, but subsequent levels will be provided by the system.
 - b. More advanced game levels will only available for highly skilled users. Those levels will be adaptively assigned as described in the workflow “M2 – Adaptive Scheduling” below.
3. If the Game Player starts playing from his/her PC, then the game will be executed on his/her computer.

Possible opportunity: to verify source code in the game-like environment on user's PC.

Possible bottleneck: there can be a performance bottleneck if there are too many simultaneously playing users – their actions can be queued and some delays can occur during game screens (according to queued data transfers).

- a. The game play will allow the user to solve some task, puzzle or to find some solution in logical model. It is expected that will be directly related to the verification of the related part of the source code for the software product under evaluation.
- b. The game task will depend on the game type and the game level.
- c. Advanced game levels will have more tasks, more difficulties and complexity than initial simple game levels.
- d. Software verification games for PC are implemented externally, so playing them is out of scope for this conceptualization.

4. If the Game Player starts playing from the mobile device, then the game will be executed on that mobile device.
Possible opportunity: to verify source code in the game-like environment on user's mobile device.
Possible bottleneck: there can be a performance bottleneck if there are too many simultaneously playing users – their actions can be queued and some delays can occur during game screens (according to queued data transfers).
 - a. The game can be played on the mobile device in the similar way as it was described above for PC game play.
 - b. The system will perform needed integration between the Gaming Website and native game played on the mobile device, so the data is properly exchanged between the mobile device and the Gaming Website web-site.
 - c. Software verification games for mobile devices are implemented externally, so playing them is out of scope for this conceptualization.
5. The Game Player will be able to participate in Tournaments, or Team based gaming events. Those competitions can have some predefined prizes/gifts or virtual awards/badges.
Possible opportunity: to support events, competition between game players; to attract users by varying the gaming formats and keep the games fun and challenging.
 - a. The Game Player can attend special contests – like short competitions on a single game with multiple other game players.
 - b. It will be possible to participate in long tournaments (usually involved many game levels or even multiple games) and team competitions.
6. The user is scored after the game level completion based on the game rules.
Possible opportunity: to provide numerical metrics of the user performance and skills for the played games.
 - a. The system will calculate user scores for the played games according to various game specific constructs and rules.
 - b. The system will calculate user placement (in comparison with other competing users) according to his/her scores.
 - c. The system will assign related awards, badges, physical prizes/gifts to the top placed users for the played games.
7. If a special achievement was earned by the Game Player, then the application will automatically notify him/her by e-mail.
Possible opportunity: to promptly notify the user about earned achievement.
 - a. The system will send the notification e-mail with brief information about the related game/contest/tournament, achieved scores, placement and type of the award.
8. The user can log out from the application to ensure high security – this application provides quite sensitive data to him/her, so the user has to logout as soon as he/she no longer needs to access the application.

M2 – Game Processing workflow and Game API:

1. The system utilizes Adaptive Scheduling for optimization of game levels to be provided for players. Adaptive scheduling itself is out of scope for this conceptualization, but the infrastructure for its integration will be provided (for calculation of users' skills, automated assignment of game levels to users according to their skills, scheduling games for users of appropriate skills, and scaling of a single game through multiple users).
2. If the game was started by the user, then the System will load chosen game and provide the optimal game level to be for the user.
Possible opportunity: to easily load the needed game components and game level from storage for the user.
 - a. The system will retrieve the game level data and use it during game play.

- b. The system will retrieve the game instance and provide it for running on the user's PC or on the user's mobile device.
3. The System will persist all the game related user performance data during game play via a standardized API.
Possible opportunity: to flexibly integrate games with the Gaming Website and seamlessly keep all the game-related data in the database.
Possible bottleneck: there can be a performance bottleneck if too many games are played simultaneously on the Gaming Website. It can produce slowdown in data persistence and game play.
 - a. The system will integrate all the games to the Gaming Website through standard interfaces, so the games are not restricted to any particular platform – i.e. mobile, browser, etc... A generic database schema will be defined and used for data storage and easy integration with any games.
 - b. All the temporary data of the game are persisted to the database. It will be possible to retrieve that data through Web Services API.
 - c. The System will store all the user achievements (including scores, placements, awards, badges, physical prizes/gifts) to the database.
 - d. The System will acquire all the game results and user optimal solutions.
 - e. The game results and optimal solutions found by the users will be stored to the database.
4. The system will persist not just level completion events, but intermediate or incomplete progress, and use the data for Adaptive Scheduling and Game feedback.
Possible opportunity: to properly determine if the levels are sufficiently designed, unsolvable, etc... This data will be valuable to the game developers and the adaptive scheduling component.

M3 – Site Administrator:

Please note, there are no bottlenecks expected in this workflow, because not many Site Administrators are expected, and they will not get a large amount of data from the application.

1. The Site Administrator will login to the Gaming Website application by providing his/her credentials.
 - a. Login will be performed via secure login or OpenID.
 - b. The application will authenticate and authorize the user according to his/her user role (i.e. Site Administrator) and provide the related functionality.
 - c. Each user will get data access in the application according to his/her user role.
2. The Site Administrator can freely manage any content on the Gaming Website.
Possible opportunity: to allow full control of all the data posted on the Gaming Website.
 - a. He/she can create new blogs, forum posts, wiki pages, write on personal walls of users.
 - b. He/she can edit any existing content entries of any users.
 - c. He/she can remove any existing content entries of any users.
3. The Site Administrator can post a news entry to the Gaming Website.
Possible opportunity: to inform users of the news related to the Gaming Website, marketing, and other promotional information.
 - a. The Site Administrator may provide regular news articles (with header, rich-formatting text and, optionally, pictures) that will be posted on the Gaming Website.
4. If the user posted any new data or changed some data on the Gaming Website, then the Site Administrator will moderate that content.
 - a. The Site Administrator will get a list (or some visually illustrated queue) of all the new/changed user content (forums, blogs, wiki pages and personal walls) and will be required to view that content.

- b. If any content is recognized as illegal or inappropriate, then the Site Administrator can freely edit or remove it.
 - c. The Site Administrator can punish users that posted illegal on inappropriate content (like by suspending their accounts for a predetermined period, or fully remove them from the application).
- 5. The Site Administrator can freely manage all the users of the Gaming Website.
Possible opportunity: to fully control all the users of the Gaming Website.
 - a. He/she can add new users, modify data of the existing users, disable/enable or even remove existing users from the application.
 - b. He/she can modify roles and/or access rights of users.
- 6. Site Administrator can directly communicate with any user.
Possible opportunity: to promptly notify given users about upcoming events from the Gaming Website.
 - a. The Site Administrator can send special announcements or notifications to the chosen users by e-mail.
 - b. The e-mail message will contain the information about the special event as free form text, provided by the Site Administrator.
- 7. Site Administrator has access to perform configuration changes.
Possible opportunity: to flexibly setup application options.
 - a. The Site Administrator can configure the application on some GUI pages, so options like e-mail server, paths to FTP servers, logging levels, etc. can be easily setup.
- 8. The Site Administrator can perform logout from the application to ensure high security – this application provides quite sensitive data to him/her, so the Site Administrator has to logout as soon as he/she no more needs to access the application.

M3 – Game Administrator:

- 1. The Game Administrator will login to the Gaming Website application by providing his/her credentials.
 - a. Login will be performed via secure login or OpenID.
 - b. The application will authenticate and authorize the user according to his/her user role (i.e. Game Administrator) and provide the related functionality.
 - c. Each user will get data access in the application according to his/her user role.
- 2. The Game Administrator can freely manage all the tools needed for conversion of software source code to game levels and for preparing verification results.
Possible opportunity: to flexibly integrate all the needed external tools for conversion of the software source code to game levels, for verification of the software code.
 - a. The Game Administrator can provide external tools for converting software product's source code to the related game levels. Those tools will be integrated into the Gaming Website.
 - b. He/she can also provide the verification and analysis tools for conversion from achieved game results and user solutions for formal software verification output. Those tools will be integrated into to the Gaming Website.
 - c. Compilers needed for preparing games can be uploaded by the Game Administrator to the Gaming Website.
 - d. All those tools and compilers will be seamlessly integrated by the system.
- 3. The Game Administrator can setup new games for the software products to be verified on the Gaming Website.
Possible opportunity: to setup games, tools, game levels on the Gaming Website.
 - a. The Game Administrator can upload the source code to the Gaming Website for conversion into levels. It will be performed on some GUI page of the application.

Possible problem: there can be a problem that software product to be evaluated is too complicated or too special to be properly verified by the Gaming Website. Or the source code can have errors, or it can be incomplete, corrupted, etc. The application has to inform the Game Administrator about any found problem with the provided source code.

Possible bottleneck: there can be a performance bottleneck if too many large software products are being simultaneously uploaded to the Gaming Website. In this case, uploading of new software products will be queued until previously started uploading processed is finished.

- b. The Game Administrator will provide all the needed tools for processing uploaded source code, for converting game results to the formal verification output, and the software verification games. The tools and games will be externally developed. The Game Administrator will simply upload them to the Gaming Website.

Possible problem: the uploaded tools can be incompatible or just partially compatible with the software product.

- c. The Game Administrator can provide the externally developed games to the Gaming Website. The games will be uploaded, integrated to the system, and then configured for verification of the software source code.
- d. The Game Administrator will select proper tools for generating needed levels for each game for software product verification and generate the needed game levels.

Possible problem: there can be a problem that automatically generated game levels are too difficult/complicated, or boring, or non-manageable, so human Game Players will not be able to complete them.

Possible problem: there can be a problem that generating game levels takes too much time. Some progress indicator is needed to inform the user about ongoing process.

Possible bottleneck: there can be a performance bottleneck if too many large software products are being simultaneously processed for game levels generation. Some generating tools can be queued until previously started generating tools are finished.

- e. The Game Administrator will ensure that the Metadata that describes each level is accurate sufficient, and compliant. This Metadata will describe each level and will be utilized by the Adaptive Scheduler.
 - f. The Game Administrator can freely configure options of the uploaded games on the related GUI pages.
- 4. If there are some game results and the Game Administrator wants to customize conversion of game results to verification output documents, then he/she perform that customization..
 - 5. The Game Administrator can perform logout from the application to ensure high security – this application provides quite sensitive data to him/her, so the Game Administrator has to logout as soon as he/she no more needs to access the application.

M3 – Software Owner:

- 1. The Software Owner will login to the Gaming Website application by providing his/her credentials.
 - a. Login will be performed via secure login or OpenID.
 - b. The application will authenticate and authorize the user according to his/her user role (i.e. Software Owner) and provide the related functionality.
 - c. Each user will get data access in the application according to his/her user role.
- 2. The Software Owner will provide the source code of his/her software product to the system.

Possible problem: a problem can be that the source code is too large, so some

progress indicator is needed during uploading to inform the user.

Possible bottleneck: there can be a performance bottleneck if too many large software products are being simultaneously uploaded to the Gaming Website. In this case, uploading of new software products will be queued until previously started uploading processes are finished.

- a. Source code of the software product can be simply uploaded by the Software Owner to the application.
 - b. That data can be placed to the application in the offline mode too (externally).
3. The Software Owner may convert the source code into game levels himself/herself for testing or for reasons such as confidentiality, etc.

Possible opportunity: to improve security for sensitive software products.

Possible problem: there can be a problem that automatically generated game levels are too difficult/complicated, or boring, or non-manageable, so human Game Players will not be able to complete them.

Possible problem: there can be a problem that generating game levels takes too much time. Some progress indicator is needed to inform the user about ongoing process.

Possible bottleneck: there can be a performance bottleneck if too many large software products are being simultaneously processed for game levels generation. Some generating tools can be queued until previously started generating tools are finished.

- a. The Software Owner can use some available conversion tools (externally developed, out of scope in this conceptualization) to convert source code to the game levels.
 - b. Generated game levels will be stored in the system and used for further verification of the software product.
4. Game Administrators will prepare needed games, tools, configuration options, etc. for the software product verification – like described in the “M3 – Game Administrator” workflow above.

Possible opportunity: to allow flexible and easily customizable usage of tools and games for software products verification.

Possible problem: the uploaded tools or games can be incompatible or just partially compatible with the software product.

5. The software product is verified by Game Players as described in the “M2 – Game Player” workflow above.

Possible opportunity: to verify source code in the game-like environment.

Possible bottleneck: there can be a performance bottleneck if there are too many simultaneously playing users – their actions can be queued and some delays can occur during game screens (according to queued data transfers).

6. The Software Owner can get verification results on his/her software product from the system.

Possible opportunity: to deliver formal verification documents for the software products to the related Software Owners.

Possible problem: there can be a problem that collected game results and user solutions are incomplete or incorrect, so the generated documents will not fully cover verification and evaluation details of the software product and can contain mistakes.

- a. The Software Owner can download the formal output of the verification results on his/her software after all games are finished on his/her product.
 - b. It will be also possible to get formal output for parts of the software product already verified by the related game(s) and game levels.
 - c. The Software Owner can download the formal output as documents from the application, or through the FTP server, or manually in the offline mode (externally).
7. The Software Owner can perform logout from the application to ensure high security – this application provides quite sensitive data to him/her, so the Software Owner has to logout as soon as he/she no more needs to access the application.

M4 - Reporter:

Please note, there are no bottlenecks expected in this workflow, because not many Reporters are expected in the application.

1. The Reporter will login to the Gaming Website application by providing his/her credentials.
 - a. Login will be performed via secure login or OpenID.
 - b. The application will authenticate and authorize the user according to his/her user role (i.e. Reporter) and provide the related functionality.
 - c. Each user will get data access in the application according to his/her user role.
2. The Reporter will view all the statistics, analytics, and verification results in real-time.

Possible opportunity: to provide full analytical and statistical data about the Gaming Website, users and played games in real-time.

 - a. Statistics, analytics, and verification output results will be dynamically displayed on a special screen.
 - b. The Reporter can view “live” charts and tables of details and summary of the Gaming Website usage, user activity, and software verification progress.
3. If the Reporter needs a prepared verification report for any given software product, then he/she can request it from the application.

Possible opportunity: to provide automatically generated full verification report to the user.

 - a. The Software Verification Report is a finalized formal output report on the verification of the software source code.
 - b. It will be a professionally formatted report with detailed and summary information about quality and found bugs of the evaluated software product.
 - c. The Reporter can get verification report (and all the other report) in one of the following formats: HTML, XML, CSV, and JSON formats.
 - d. The application will display the report on the screen.
4. If the Reporter needs more reports, then he/she can get all the specific reports from the application.

Possible opportunity: to provide various useful reports about the application to the user.

 - a. The Reporter will select the type of the needed report – reports on the usage of the application by various users, most popular pages and content will be provided too. The user statistics, preferences, and achievements report will be provided as well. The report on game results, scores, found optimal solutions, and other game statistics will be available too.
 - b. The chosen report will be displayed on the screen in one of the supported formats (HTML, XML, CSV, and JSON).
5. If the Reporter needs customization of reports, then he/she can easily perform that in the application.

Possible opportunity: to allow full customization of reports by the user.

 - a. The Reporter can choose needed parts of the report, needed columns, data sorting, filtering, etc.
 - b. The customized report will be displayed on the screen in one of the supported formats (HTML, XML, CSV, and JSON).
6. If the Reporter needs to receive scheduled reports through some FTP server, then he/she can ask the application about that.

Possible opportunity: to allow automated regular publish of reports on the remote FTP server.

 - a. The user can choose the needed schedule for generating needed reports.
 - b. The application will automatically generated selected reports by the schedule and persist them in the system.

- c. The Reporter will be able to download the scheduled report from the FTP server.
7. If the Reporter needs to export the displayed report, then he/she can ask the application to provide the report file.
Possible opportunity: to provide reports as files for further usage outside the application.
 - a. The application will allow exporting any of the displayed reports in one of the following formats: HTML, XML, CSV, JSON, PDF and TXT.
 - b. The report will be exported to a downloadable file, so the user can download it by his/her web-browser to the local file system.
8. If the Reporter needs to make a printed copy of the displayed report, then he/she can easily do that in the application.
Possible opportunity: to easily get a “hardcopy” of any report.
 - a. Any displayed report can be freely printed by the Reporter.
 - b. The application will print the current web-page of the report by user’s web-browser.
 - c. It will be possible to print a chosen table on multiple pages, so full large reports can be easily printed.
9. The Reporter can perform logout from the application to ensure high security – this application provides quite sensitive data to him/her, so the Reporter has to logout as soon as he/she no more needs to access the application.

M5 – Generating LiveDVD:

1. If the Administrator needs to use some new distribution of the OS, then he/she can upload it to the system.
Possible opportunity: to easily upload needed OS for LiveDVD generation.
Possible problem: there can be a problem that OS distribution is too large, so some progress indicator is needed during uploading to inform the user.
Possible bottleneck: there can be a performance bottleneck if too many large distributions are being simultaneously uploaded to the application. In this case, uploading of new distributions will be queued until previously started uploading processes are finished.
 - a. All the uploaded OS distributions are kept and can be used later for generating of LiveDVD.
2. If the Administrator needs to use some new distribution of the Gaming Website, then he/she can upload it to the system.
Possible opportunity: to easily upload needed Gaming Website for LiveDVD generation.
Possible problem: there can be a problem that Gaming Website distribution is too large, so some progress indicator is needed during uploading to inform the user.
Possible bottleneck: there can be a performance bottleneck if too many large distributions are being simultaneously uploaded to the application. In this case, uploading of new distributions will be queued until previously started uploading processes are finished.
 - a. All the uploaded Gaming Website distributions are kept and can be used later for generating of LiveDVD.
3. If the Administrator has some new external tools, then he/she can upload them to the system.
Possible opportunity: to easily upload needed external tools for LiveDVD generation.
 - a. All the uploaded external tools are kept and can be used later for generating of LiveDVD.
4. If the Administrator has some new games, then he/she can upload them to the system.
Possible opportunity: to easily upload needed games for LiveDVD generation.

- a. All the uploaded games are kept and can be used later for generating of LiveDVD.
5. If the Administrator needs to use some new user guides or troubleshooting guides (documents), then he/she can upload them to the system.
Possible opportunity: to easily upload needed documents for LiveDVD generation.
 - a. All the uploaded documents are kept and can be used later for generating of LiveDVD.
6. The Administrator can select any needed OS distribution from the set of already uploaded ones for further generating of the LiveDVD.
7. The Administrator can select any needed Gaming Website distribution from the set of already uploaded ones for further generating of the LiveDVD.
8. The Administrator can select any needed external tool distributions from the set of already uploaded ones for further generating of the LiveDVD.
9. The Administrator can select any needed game distributions from the set of already uploaded ones for further generating of the LiveDVD.
10. The Administrator can select any needed user guide distribution and user troubleshooting guide distributions from the set of already uploaded ones for further generating of the LiveDVD.
11. The Administrator will automatically generate the LiveDVD with the chosen OS distribution, Gaming Website distribution, external tool distributions, game distributions, user guide and troubleshooting guide distributions.
Possible opportunity: to flexibly prepare working bootable LiveDVD from the chosen distributions.
Possible problem: there can be a problem that LiveDVD generation process is too long, so some progress indicator is needed during generating to inform the user.
Possible bottleneck: there can be a performance bottleneck if too many LiveDVDs are being simultaneously generated by the application. In this case, generating of new LiveDVDs will be queued until previously started generating processes are finished.
 - a. The image of the LiveDVD can be created by some backend scripts. It will be downloadable by the Site Administrator.
 - b. The LiveDVD will be a bootable CD or DVD disk, which can be used for full installation of the Gaming Website of the customer's site.
 - c. The LiveDVD may also used to boot and immediately run the Gaming Website instance right from that CD or DVD disk depending on hardware requirements.
12. The Administrator can use the prepared LiveDVD as follows:
Possible opportunity: to easily install (or just run) the private instance of the Gaming Website.
Possible problem: there can be a problem that generated LiveDVD isn't compatible with the user's hardware or has other problems during installation of the Gaming Website.
 - a. The private instance can be installed from the LiveDVD to the customer's computer and then it will be used as a private instance of the Gaming Website.
 - b. It will be also possible to simply runt the private instance of the Gaming Website directly from the LiveDVD (without installation).
13. The operating system and private Gaming Website will be started from LiveDVD and the user can use the prepared private instance like a local (or offline) Gaming Website for performing private competitions within the customer's site.
Possible opportunity: to allow usage of the private instance of Gaming Website in the similar way like the public Gaming Website, but only by the limited set of users on the customer's site.
Possible problem: there can be a problem that generated LiveDVD isn't compatible with the user's hardware or has other problems during running the Gaming Website.

- a. All the actions of regular Gaming Website (like described in modules M1 – M4) can be performed on the privately used instance of Gaming Website and the data (like software products, game levels, formal verification results, etc.) can be freely interchanged between the public Gaming Website and the private Gaming Website according to the needs of private Gaming Website instance owners.
14. The data between public and private instances of the Gaming Website can be transferred online (through some API), or in offline mode (through some storage media).
Possible opportunity: to allow easy communication between public Gaming Website and private Gaming Website instances.
Possible problem: there can be a problem that private Gaming Website instance is too strictly protected by firewall and can't communicate with the publish Gaming Website. Usage of some offline data storage media can be used in this case to manually transfer the needed data between Gaming Websites.




Services APIs:

1. The external sub-system (like a conversion tool, game, etc.) can get programmatic access to the functionality of the application (authentication is required).
 - a. The application will provide Web Service API, accessible through OAuth authentication.
2. The external sub-system will get access to all the web services' API after successful authentication.
3. The external sub-system can choose any of the available APIs and programmatically access the related functionality of the Gaming Website:
Possible opportunity: to allow easy integration of external sub-systems with the Gaming Website.
Possible problem: there can be a problem that the network connections with web services can be lost – in this case API of the application will be not accessible.
Possible bottleneck: there can be a performance bottleneck if too many external sub-systems are simultaneously accessing the application's API. In this case, new requests to web services will be queued until all the previous requests are processed.
 - a. It will be possible to get access to all the gaming functionality through the Game Web Services API. Game levels and results will be transmitted as well through that API.
 - b. It will be possible to get access to all the administrative functionality through the Admin Tools Web Services API.
 - c. It will be possible to get access to all the reports, "live" statistics, and analytics data of the software verification process and Game Service usage through the Reporting Web Services API.
4. The external sub-system can disconnect from the application's web services and stop using them when needed.

3.4 Use Case Diagrams

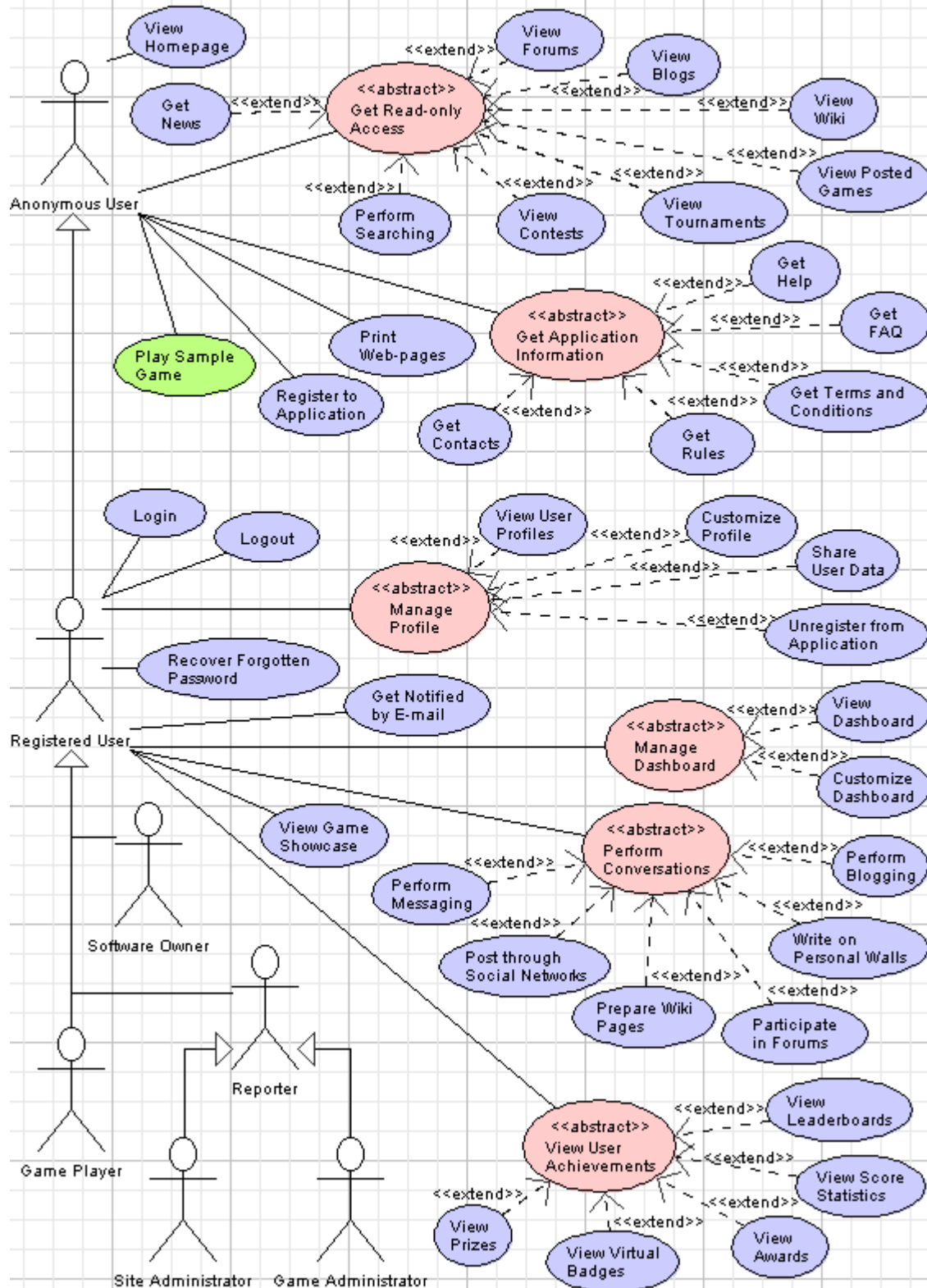
Use case diagrams are helpful illustrations of how the application's users interact with the system. Use case "bubbles" correlate to distinct task domains and are useful in determining user roles, module boundaries, shared tasks, etc. The use case diagrams are intentionally high level, and exclude activity details such as for example "submit shopping cart." The diagram below should be used to verify that the required functionality of the application is covered.

The use case diagrams use the next coloring schema for bubbles:

Color	Description
	The regular (concrete) use case
	The aggregate use case
	The external use case (not in scope)

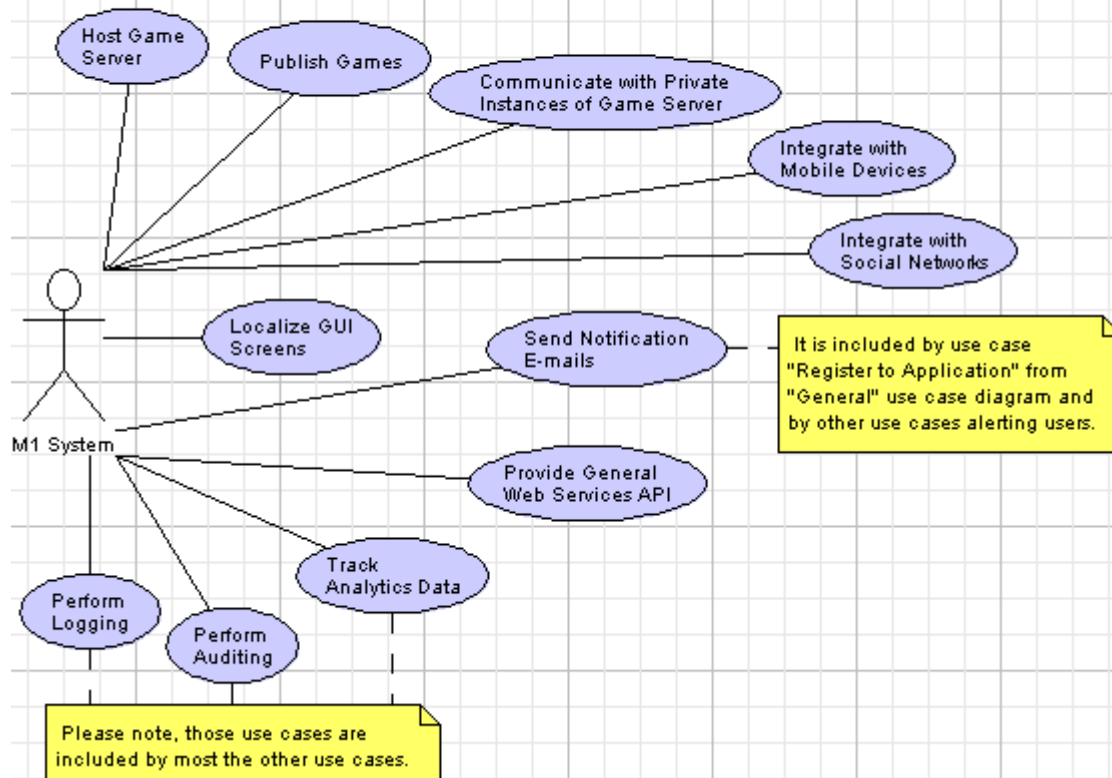
Gaming Website 1.0 - (M1, General) Use Case Diagram

M1 (Gaming Community Website) module of CSFV Community Gaming Website 1.0 application.



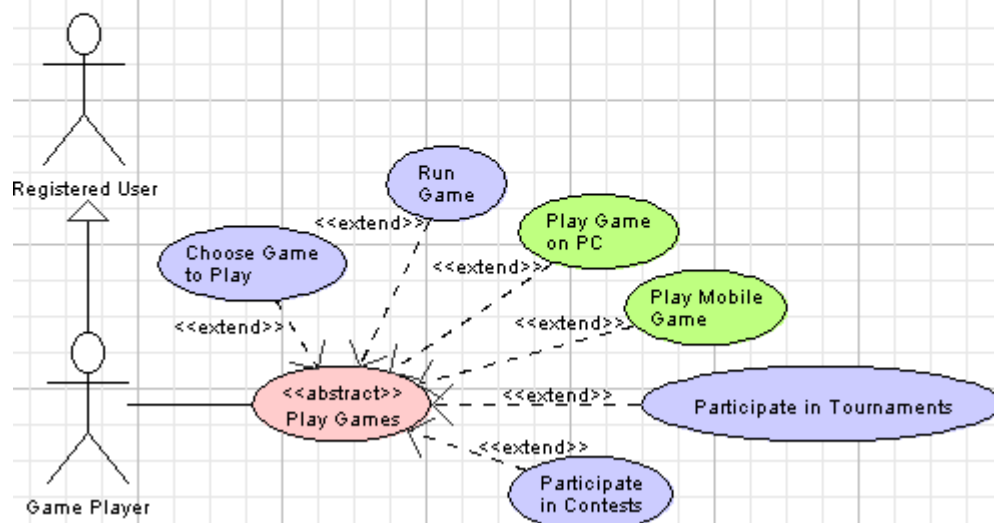
Gaming Website 1.0 - (M1, System) Use Case Diagram

M1 (Gaming Community Website) module of CSFV Community Gaming Website 1.0 application.



Gaming Website 1.0 - (M2, Game Player) Use Case Diagram

M2 (Gaming) module of CSFV Community Gaming Website 1.0 application.



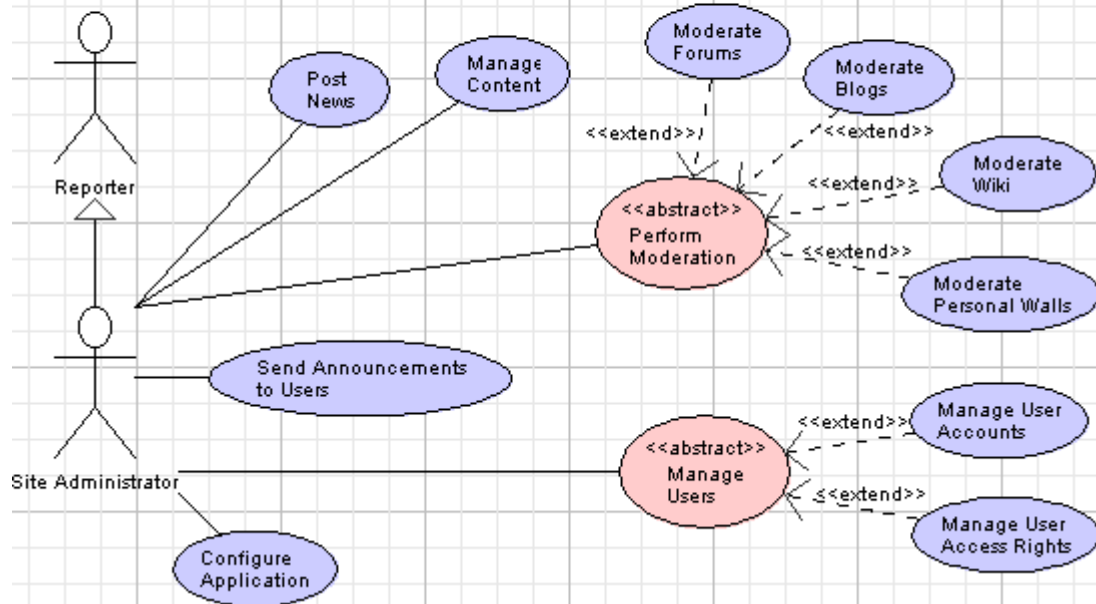
Gaming Website 1.0 - (M2, System) Use Case Diagram

M2 (Gaming) module of CSFV Community Gaming Website 1.0 application.



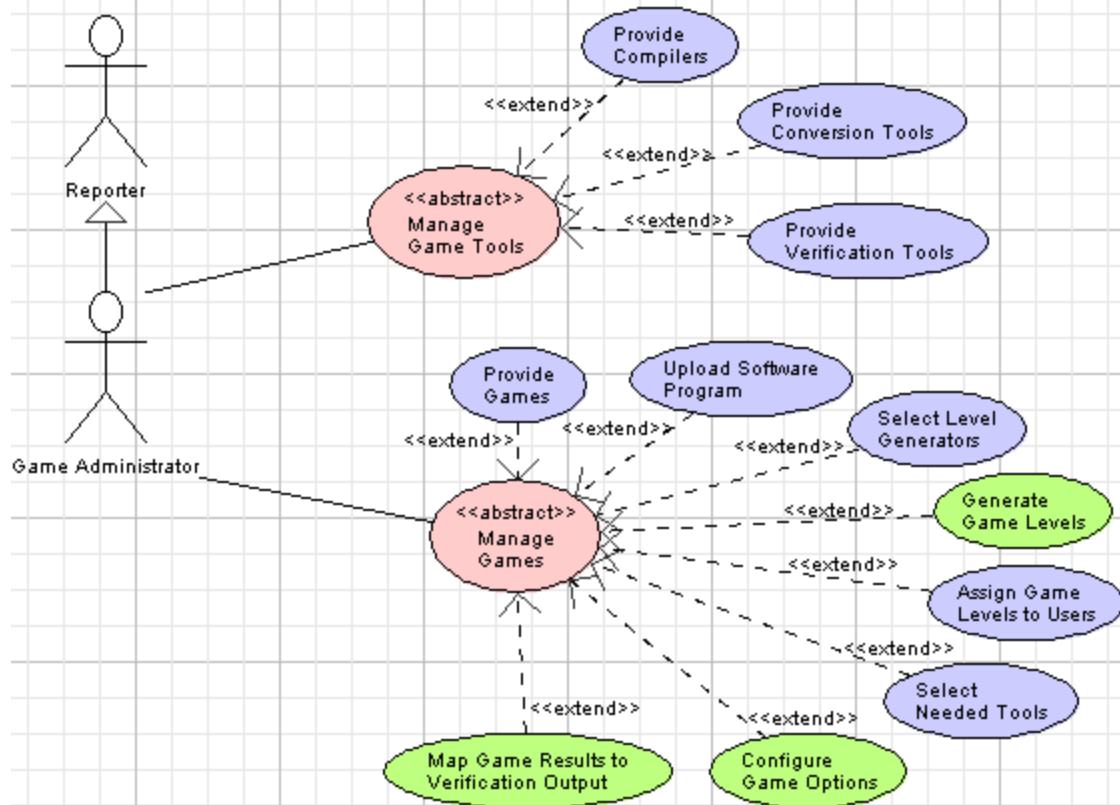
Gaming Website 1.0 - (M3, Site Admin) Use Case Diagram

M3 (Admin tools) module of CSFV Community Gaming Website 1.0 application.



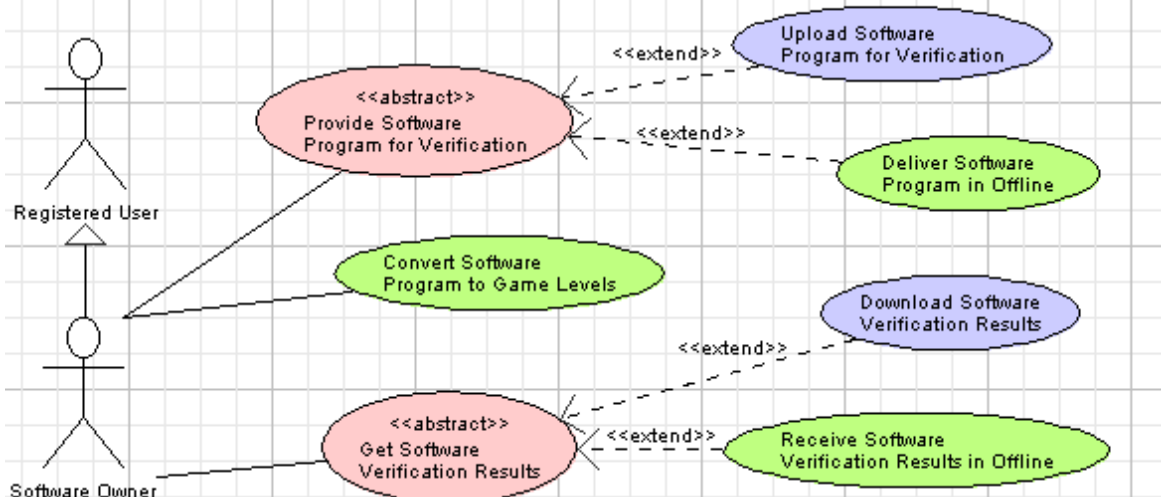
Gaming Website 1.0 - (M3, Game Admin) Use Case Diagram

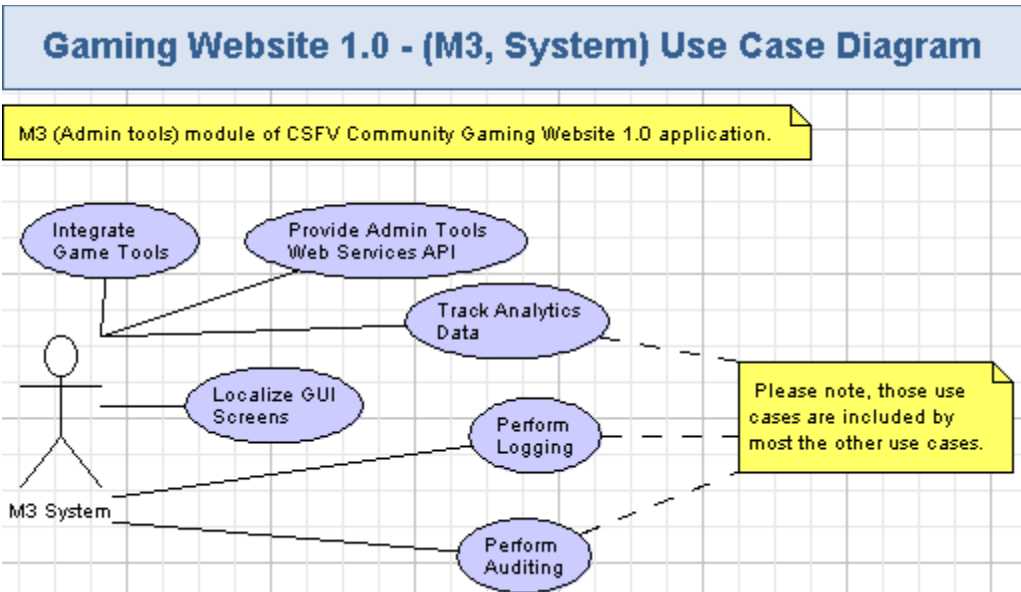
M3 (Admin tools) module of CSFV Community Gaming Website 1.0 application.



Gaming Website 1.0 - (M3, Software Owner) Use Case Diagram

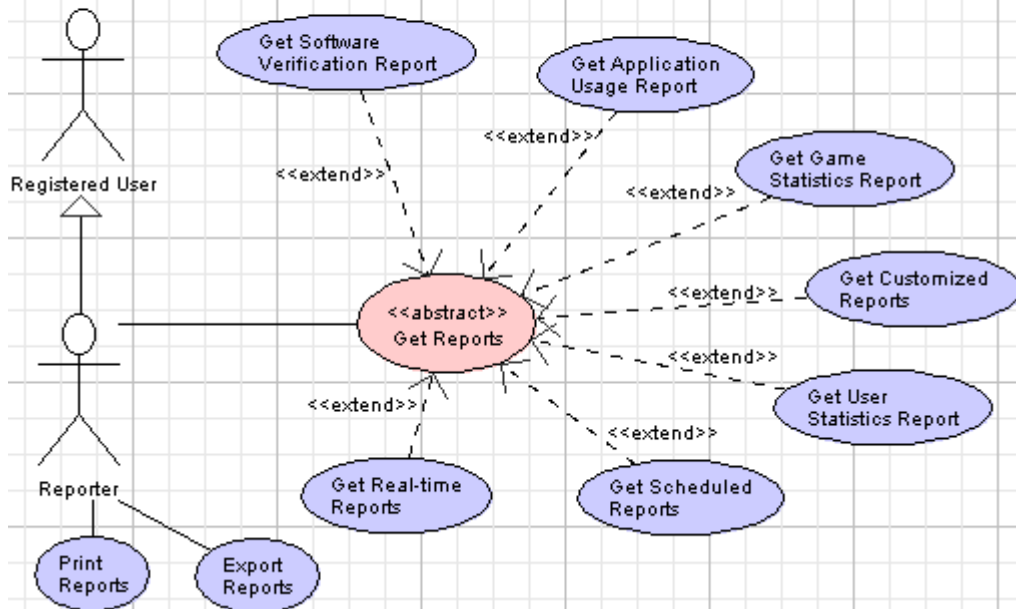
M3 (Admin tools) module of CSFV Community Gaming Website 1.0 application.





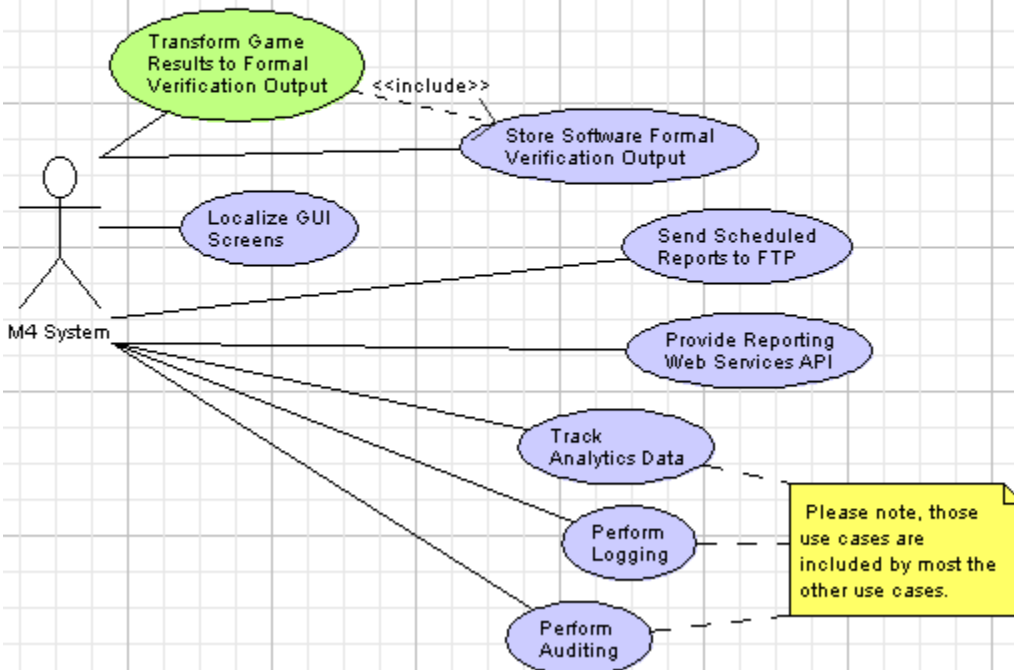
Gaming Website 1.0 - (M4, Reporter) Use Case Diagram

M4 (Reports) module of CSFV Community Gaming Website 1.0 application.



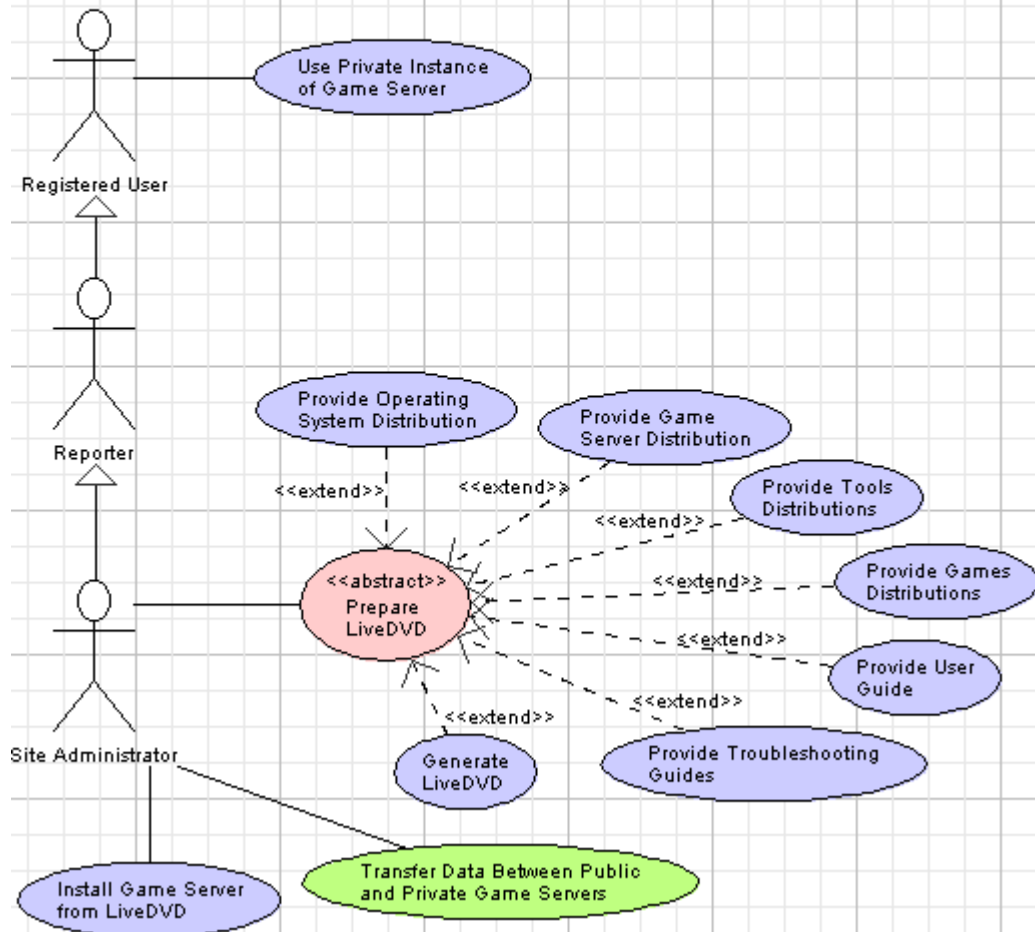
Gaming Website 1.0 - (M4, System) Use Case Diagram

M4 (Reports) module of CSFV Community Gaming Website 1.0 application.



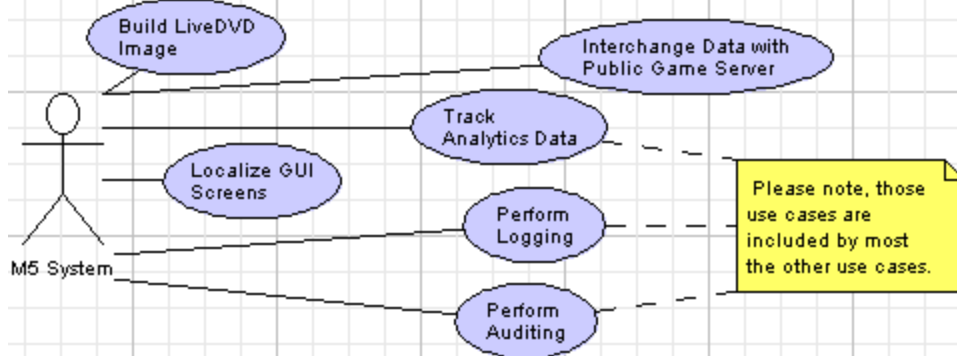
Gaming Website 1.0 - (M5, LiveDVD) Use Case Diagram

M5 (Build and Installation DVD) module of CSFV Community Gaming Website 1.0 application.

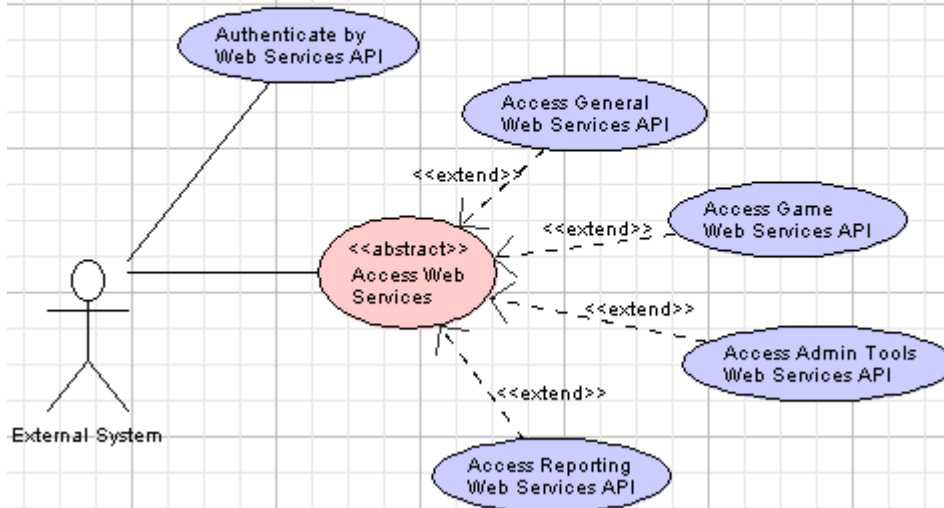


Gaming Website 1.0 - (M5, System) Use Case Diagram

M5 (Build and Installation DVD) module of CSFV Community Gaming Website 1.0 application.



Gaming Website 1.0 - (External System) Use Case Diagram



4.0 Business Requirements

This section identifies, enumerates and explores the business requirements that must be met by the application. Business requirements include capturing the types of users, the basic inputs and outputs, the system's dependencies, and the tasks the system should accomplish. Some business requirement for a web-based retail site might include:

Users: "The system supports public browsers, registered members, customer service administrators, and system administrators."

Tasks: "Users have personalized accounts," "shopping carts expire in a configurable period, and the default is 10 days", etc.

It is important to confirm that Task Requirements include all the tasks required to meet the business objectives.

4.1 Users

All applications have users and most have several users of different types. This section identifies, at a high level, the *types* of users of the system.

4.1.1 *Anonymous User*

This is a user, which did not register to the application. He/she has just a read-only access to the application features. He/she can view only publicly available content and play only a few sample games. He/she can register to the application to get access to more functionality and ability to post data on the Gaming Website.

4.1.2 *Registered User*

This user role describes additional allowed features for the users after they register to the application. The registered user can perform all actions of the non-registered user and have many edit/write abilities in the application (including posting content to the Gaming Website, communicating with other community users, viewing user achievements, viewing profiles of other users and managing his/her own profile).

4.1.3 *Game Player*

This is the user playing software verification games. This is a registered user and additionally he/she can choose needed game, run and play it either on his/her PC or on the mobile device, participate in short contests, in long tournaments (and team competitions).

4.1.4 *Site Administrator*

This is a management person controlling usage of the application. This is a registered user and additionally he/she can post news, manage content, moderate data posted by the user, manage all the users (accounts and their user roles), configure Gaming Website. Site Administrator can create LiveDVD disks and install private instances of the Gaming Website. This user role has also all functionality of the Reporter user role.

4.1.5 *Game Administrator*

This is a management person managing games. This is a registered user and additionally he/she can upload software programs, tools, and games, select needed tools, configure games, generate game levels and assign them to the related users, customize conversion of game results and found user solutions to the formal verification output documents. This user role has also all functionality of the Reporter user role.

4.1.6 *Software Owner*

This is the customer having software products to be verified on the Gaming Website. This is a registered user and additionally he/she can upload software source code to the Gaming Website (in online or offline mode), convert source code to game levels, and to download finally generated formal verification output documents for his/her software products from the Gaming Website (in online or offline mode).

4.1.7 *Reporter*

This is a special reporting user role. The Reporter can obtain real-time reports with the application statistics, analytics, user activity, and verification results information. He/she can get verification reports for the software products, customize them, and get scheduled reports (from some FTP server), export and print reports.

4.1.8 *External System*

This is not a human user role, but some system which needs to integrate with the Gaming Website. It can authenticate to the Web Services API of the Gaming Website, and get programmatic access to all the general, gaming, administration, and reporting functionality of the Gaming Website.

An M1 System actor was added to the use case diagram to specially highlight that logging, auditing, tracking user actions and data modifications, hosting Gaming Website, publishing games, integrating with mobile devices and social networks, sending e-mail notifications, localizing GUI screens, and providing general web services API are automatically performed by the module M1 of the application (like in the background).

An M2 System actor was added to the use case diagram to specially highlight that logging, auditing, tracking user actions and data modifications, integrating games and adaptive scheduling, processing game data, calculating scores and achievements, localizing GUI screens, and providing game web services API are automatically performed by the module M2 of the application (like in the background).

An M3 System actor was added to the use case diagram to specially highlight that logging, auditing, tracking user actions and data modifications, integrating game tools, localizing GUI screens, and providing administration tools' web services API are automatically performed by the module M3 of the application (like in the background).

An M4 System actor was added to the use case diagram to specially highlight that logging, auditing, tracking user actions and data modifications, generating of verification output, sending scheduled reports by FTP, localizing GUI screens, and providing reporting web services API are automatically performed by the module M4 of the application (like in the background).

An M5 System actor was added to the use case diagram to specially highlight that logging, auditing, tracking user actions and data modifications, interchanging data between private and public instances of the Gaming Website, and localizing GUI screens are automatically performed by the module M5 of the application (like in the background).

4.2 **Inputs/Outputs**

This section identifies and describes the inputs to and outputs from the new application. Inputs can include electronic inputs, like RSS and EDI feeds, updates from external databases, etc, as well as human inputs, like "user X keys in results from report Y." Output can include electronic feeds, printed reports, etc. In this section, all electronic inputs and outputs are captured. Human inputs are omitted from this section.

4.2.1 Inputs

4.2.1.1 Software Products Source Code

The Software Owner or Game Administrator can upload the source code of the software product to be verified. Providing source code can be performed either in the online or offline mode.

4.2.1.2 Games

The externally developed games will be uploaded to the application for further usage during software products verification.

4.2.1.3 Tools

The externally developed tools (and compilers) will be uploaded to the application for further usage for game levels generation and converting game results and user solutions to the formal output documents.

4.2.1.4 Login Credentials

The user has to provide his/her credentials (user name and password) to login to the application.

4.2.1.5 Authentication to Access Web Services API

The external system will authenticate with OAuth to get access to the Web Services API, provided by the Gaming Website.

4.2.1.6 Search Queries

The user can enter search queries to find needed games, tools, contests, tournaments, user profiles, forum threads, blog posts, wiki pages, etc. in the application. Basic and advanced searching criterions will be supported.

4.2.1.7 Application Configuration Data

The Site Administrator will enter/change setup options to the application – like through some GUI pages of the application.

4.2.1.8 Games Configuration Data

The Game Administrator can enter/change setup options for the externally provided games – like through some GUI pages of those games.

4.2.1.9 User Accounts and Personal Information Management

The user can register his/her account on the application. The user can provide just a minimal set of the personal data and the new account will be created. He/she can freely edit and add more data for his/her account. Site Administrators can freely create/edit/remove accounts of all the other users.

4.2.1.10 User Access Rights Management

The administrator can assign more user roles or remove existing user roles for any user in the system.

4.2.1.11 User Accounts from Facebook, Twitter

The application can re-use existing accounts of the users from the following web-sites: Facebook, Twitter. So, the application will perform authorization and authentication through that web-site.

4.2.1.12 Forums Management and Moderation

Registered users can create new threads, edit own posts, and reply to existing posts on the forums. Site Administrators can freely moderate all the forums in the system.

4.2.1.13 Blogs Management and Moderation

Registered users can create new blog articles, edit them and comment blog articles as well. Site Administrators can freely moderate all the blogs in the system.

4.2.1.14 Wiki Management and Moderation

Registered users can create new wiki pages, edit them and comment wiki pages as well. Site Administrators can freely moderate all the wiki pages in the system.

4.2.1.15 Personal Walls Management and Moderation

Registered users can post on personal walls of community members and edit their posts. Site Administrators can freely moderate all the posts on all the personal walls in the system.

4.2.1.16 LiveDVD Disk

The Site Administrator can install or simply run a private instance of Gaming Website through the prepared LiveDVD disk.

4.2.1.17 Dashboard Customization

The user can manually customize his/her individual dashboard (like change layout, move or resize widgets, add or remove widgets).

4.2.1.18 News Management

Site Administrators can freely create and edit news articles and publish them on the Gaming Website.

4.2.1.19 Requesting Password Recovery

The registered user can request password recovery e-mail if he/she forgot the password.

4.2.1.20 Data Sharing Management

The registered user can manually choose which data of his/her profile and own content will be shared with the other community members.

4.2.1.21 Requesting Posting through Social Networks

The registered user can request the application to post some of its content through popular social networks (like Facebook, Twitter).

4.2.1.22 Messages for/from Community Members

The registered user can enter text chat message to be delivered to the related community members. He/she can also receive such messages from other community members.

4.2.1.23 Responses from other Gaming Website Instance

The application will allow integration of public and private instances of the Gaming Website through some API. The responses from other instance will be inputted to the application.

4.2.1.24 Responses from Mobile Device

The application can get response from native game applications played on the mobile devices.

4.2.1.25 Selecting the Game and Game Level

The user can choose which game to play on the Gaming Website. And it will be possible to manually select some initial game levels where to play the game.

4.2.1.26 Running the Game

The user will run the chosen game from the Gaming Website and it will be played on his/her computer or mobile device.

4.2.1.27 Playing Games [externally]

The user will play externally developed games for performing formal verification of the chosen software products.

4.2.1.28 Attending Contests

The registered user can participate in short competitions – contests. He/she will register for the contest and play the related game and game levels.

4.2.1.29 Attending Tournaments

The registered user can participate in long competitions – tournaments (and team competitions). He/she will register to the tournament and play the related games and game levels.

4.2.1.30 Selecting Tools for Software Product Verification

The administrator can choose needed tools (from the uploaded ones) for usage during verification of the related software product.

4.2.1.31 Selecting Game Levels Generators

The administrator will choose needed game level generators (tools) from the uploaded ones for generate appropriate game levels according to the software product source code.

4.2.1.32 Assigning Game Levels to Users (manual)

The game administrator can manually assign game levels to the related users (game players).

4.2.1.33 Requesting to Map Game Results and Solutions to Formal Verification Output

The Game Administrator will request the application to customize converting the available results and user solutions (of the played games) to the related formal output documents.

4.2.1.34 OS Distribution for LiveDVD

The Site Administrator can upload OS distribution to the application for further usage on the LiveDVD.

4.2.1.35 Gaming Website Distribution for LiveDVD

The Site Administrator can upload Gaming Website distribution to the application for further usage on the LiveDVD.

4.2.1.36 Tools Distributions for LiveDVD

The Site Administrator can upload tools distributions to the application for further usage on the LiveDVD.

4.2.1.37 Games Distributions for LiveDVD

The Site Administrator can upload games distributions to the application for further usage on the LiveDVD.

4.2.1.38 User Guide Distribution for LiveDVD

The Site Administrator can upload user guide distribution to the application for further usage on the LiveDVD.

4.2.1.39 Troubleshooting Guides Distributions for LiveDVD

The Site Administrator can upload troubleshooting guides to the application for further usage on the LiveDVD.

4.2.1.40 Requesting for LiveDVD Generation

The Site Administrator can manually choose needed distributions (from uploaded ones) and request the application to generate the LiveDVD with those distributions.

4.2.1.41 Requests to Generate Game Levels

Game Administrators and Software Owners can request the application to convert software source code to the related game levels by chosen conversion tools and compilers.

4.2.1.42 Reports Customization

The Reporter can freely customize all the output reports by choosing needed data parts, columns, data ordering, filtering, and formatting.

4.2.1.43 Reports Schedule

The Reporter can choose reports for automatic regular generation and setup schedule for them. The application will input that schedule and automatically prepare requested reports and send them to some FTP server according to the schedule.

4.2.1.44 Programmatic Requests to Web Services API from External Systems

The external system can perform various requests to General, Game, Admin Tools, and Reporting Web Services API of the Gaming Website. It will be possible to fully integrate external systems with the Gaming Website through those APIs.

4.2.1.45 Adaptively Assigned Game Levels

The application will receive adaptively assigned game levels from the Adaptive Scheduling sub-system.

4.2.2 Outputs

4.2.2.1 Game Results

The user will solve games' puzzles, logical tasks, etc. and that data will be collected as game results. Formal verification output documents will be generated later according to that data.

4.2.2.2 User Optimal Solutions

The user can optimize some models, diagrams, etc. during game play and that data will be collected as user optimal solutions. Formal verification output documents will be generated later according to that data.

4.2.2.3 Formal Verification Output Documents and Reports

The Software Owner or Reporter can get output documents with software verification formal results and reports – it can be performed either in the online or offline mode. Reports will be displayed in HTML, XML, JSON, and CSV formats.

4.2.2.4 Application Usage Report

The system will display the report on the application usage to the administrator, so he/she can analyze how efficient the application was used. Reports will be displayed in HTML, XML, JSON, and CSV formats.

4.2.2.5 Game Statistics Report

The system will display the report on game statistics to the administrator, so he/she can analyze most popular, weak and bottleneck parts of the games. Reports will be displayed in HTML, XML, JSON, and CSV formats.

4.2.2.6 User Statistics Report

The system will display the report on user statistics to the administrator, so he/she can analyze involvement of users, their abilities, preferences, etc. Reports will be displayed in HTML, XML, JSON, and CSV formats.

4.2.2.7 Exported Reports

The Software Owner or Reporter can export any displayed report. Reports will be exported in HTML, XML, JSON, CSV, PDF, and TXT formats.

4.2.2.8 Real-time Reports

The Software Owner or Reporter can get all the statistics, analytics, and verification results data in the real-time – like through dynamically updated “live” pages.

4.2.2.9 Scheduled Reports

The Software Owner or Reporter can get all the scheduled reports that were automatically generated and sent to some FTP server.

4.2.2.10 Customized Reports

The Software Owner or Reporter can get the customized reports (i.e. reports with flexibly chosen data parts, columns, data ordering, filtering and formatting). Reports will be exported in HTML, XML, JSON, CSV, PDF, and TXT formats.

4.2.2.11 Generating LiveDVD

The application will generate a LiveDVD disk according to the provided and chosen distributions. The disk will be bootable, so the user can easily install or simply run the distributions software. That disk can be user later for installing or running private instance of the Gaming Website. The disk image (either for CD or DVD) will be downloadable by the Site Administrator.

4.2.2.12 Private Gaming Website Instance

The application will enable use of a private version of Gaming Website using the LiveDVD.

4.2.2.13 Found Results

The application will find content (games, tools, contests, tournaments, user profiles, etc.) according to the performed search queries and show found results.

4.2.2.14 Help Data

The application will provide help information on its usage (guides, getting started, troubleshooting guides, etc.) to the user.

4.2.2.15 Printed Web-pages and Printed Reports

The user can get hard copy printout of any web-page of the application (including various reports).

4.2.2.16 Reset User Password E-mail

The system will send user account password reset e-mail to the related user on his/her request. That e-mail will allow the user to enter new password for his/her account if the old password was forgotten.

4.2.2.17 User Dashboard

The application will provide a customized web-page, which fetches data from various parts of the application, to each registered user.

4.2.2.18 Displaying Forum Threads and Posts

The application will show forum threads and posts to the users of Gaming Website.

4.2.2.19 Displaying Blog Posts and Comments

The application will show blog posts and comments to the users of Gaming Website.

4.2.2.20 Displaying Wiki Articles and Comments

The application will show wiki articles and comments to the users of Gaming Website.

4.2.2.21 Displaying Personal Wall Posts

The application will show posts on personal walls to the registered users of Gaming Website.

4.2.2.22 Displaying User Profiles

The application will display community member profiles to the registered users of Gaming Website.

4.2.2.23 Displaying User Roles

The Site Administrators can freely view user roles of community members.

4.2.2.24 Notification E-mails

The application will send various notification e-mails to users (like on the account registration, on some achievements, on announcements, etc.).

4.2.2.25 Data Sharing to Social Network

The application will allow sharing of data from the Gaming Website with social networks (like Facebook, Twitter).

4.2.2.26 Displaying News

The application will show news articles to the users of Gaming Website.

4.2.2.27 Displaying Posted Games

The application will show the list of currently posted games on the Gaming Website.

4.2.2.28 Displaying Contests

The application will show the currently available short contests on the Gaming Website.

4.2.2.29 Displaying Tournaments

The application will show the currently available long tournaments (and team competitions) on the Gaming Website.

4.2.2.30 Logging Data

The application will log all errors, exception, warnings and debug information to the logging system (like files on the server side).

4.2.2.31 Auditing Data

The application will audit information about all the users' actions and created/edited/removed data records to the database.

4.2.2.32 Tracking Analytics Data

The application will track all the statistics data about the application, users, and played games to the database for further analysis and creating of the analytical reports.

4.2.2.33 FAQ Data

The application will display the page with the most frequently questions and the related answers to the users of Gaming Website.

4.2.2.34 Terms and Conditions Data

The application will display the regular Terms & Conditions page with legal information on the Gaming Website usage to the user of Gaming Website.

4.2.2.35 Rules Information

The application will show the page with rules of the Gaming Website, common rules in games, tips and hints data to the users of Gaming Website.

4.2.2.36 Contacts Data

The application will display the page with web-site owner and support team contacts information to the users of Gaming Website.

4.2.2.37 Displaying Chat Window

The application will output all the text chat messages on a special chat window (shown on the user's dashboard).

4.2.2.38 Displaying Game Showcases

The application will display detailed description of the chosen game, instructions, screenshots, etc. data on the game showcase page for the users of Gaming Website.

4.2.2.39 Leaderboards

The application will display the leaderboards with top scored community members on the Gaming Website. There can be multiple leaderboards – according to game topics, and a single summary leaderboard.

4.2.2.40 User Scores and Placements for Played Games

The application will display achieved scores and placements for the games, played by the users.

4.2.2.41 Awards

The application will display all the achieved awards to the registered users of Gaming Website.

4.2.2.42 Virtual Badges

The application will display virtual badges (like re-usable pictures) to the registered users of Gaming Website.

4.2.2.43 Physical Prizes and Gifts

The application will display the earned physical prizes and gifts to the related user. Other users can view that data only if the prize/gift owner allowed sharing of that data with the community.

4.2.2.44 Hosting Public Gaming Website

The system will host public instance of Gaming Website on the Internet, so it will be publicly available to all the Internet users.

4.2.2.45 Publishing Games

The system will automatically integrate and publish externally developed games on software verification process to the Gaming Website.

4.2.2.46 Publishing Tools

The system will automatically integrate and publish externally developed conversion and verification tools (and compiler) to the Gaming Website.

4.2.2.47 Generated Game Levels

The application will generate game levels for software source code by using externally developed tools and compilers.

4.2.2.48 Processing Software Formal Verification Workflow

The application will show steps of the formal verification workflow of the software product through the pages of the Gaming Website. And that workflow will be followed by the application.

4.2.2.49 Requests to other Gaming Website Instance

The application will allow integration of public and private instances of the Gaming Website through some API (like through FTP). The requests to other instance will be outputted from the application.

4.2.2.50 Requests to Mobile Device

The application can send requests to native game applications played on the mobile devices.

4.2.2.51 Integration of Adaptive Assignment of Game Levels to Skilled Users and Games Scheduling

The system will integrate adaptive scheduling of games and levels according to actual skills of users. Adaptive scheduling itself will be implemented externally and it is out of scope for this conceptualization. The system will send data requests with user information to the Adaptive Scheduling sub-system and receive adaptively assigned game levels for that user.

4.2.2.52 Programmatic Response from Web Services API to External Systems

The external system can receive responses (all the persisted data) from General, Game, Admin Tools, and Reporting Web Services API of the Gaming Website according to the performed requests. It will be possible to fully integrate external systems with the Gaming Website through those APIs.

4.3 Dependencies

Human inputs are also categorized as dependencies. List all dependencies.

4.3.1 *Forum*

This is a system providing standard forum functionality. Some open source software is needed for that and has to be chosen by the Architect.

4.3.2 *Blog*

This is a system providing standard blog functionality. Some open source software is needed for that and has to be chosen by the Architect.

4.3.3 *Wiki*

This is a system providing standard wiki functionality. Some open source software is needed for that and has to be chosen by the Architect.

4.3.4 *CMS*

This is a Content Management System (like Liferay, Joomla, Drupal, Magnolia), which can be used to prepare/manage various content on the Gaming Website, user profiles, personal walls, etc.

4.3.5 *OpenID*

The application will authenticate users through some own Security System, which will be integrated with OpenID.

4.3.6 OAuth

The application will authorize users through some own Security System, which will be integrated with OAuth.

4.3.7 Mobile Device

Software verification games can be played on the mobile devices. Concrete types of the mobile devices to be supported are not defined yet.

4.3.8 PC

Software verification games can be played on the regular computer (MS Windows based, Linux or MAC computer).

4.3.9 Games

The games will be used for software verification in the informal way. They will be externally implemented. Games will be uploaded to the Gaming Website and can be played for software verification of the chosen software products.

4.3.10 Tools

The tools (and compilers) will be used for generating game levels and converting game results and found solutions to the formal verification output documents. They will be externally implemented. Tools will be uploaded to the Gaming Website and can be used during software verification workflow of the chosen software products.

4.3.11 Adaptive Scheduling

The application will use externally implemented Adaptive Scheduling for automatically assign game levels to the users with appropriate skills, experience, and performance.

4.3.12 Printer

The application will be able to print any web-page (including reports) on the printer.

4.3.13 Database

The application stores user accounts, news, forum threads, blogs, wiki, games, tools, distributions, contests, tournaments, help, FAQ, terms & conditions, rules, contacts, personal walls, chat text messages, scores, leaderboards, awards, badges, prizes, placements, skills, software products, game results and user solutions, formal verification output, game level assignments, game schedules, game configurations, Gaming Website configuration, tools selections, report schedules and customization. And it will audit user actions and data modifications. It will track statistics data.

4.3.14 Web Server

The application will provide its GUI through the web server.

4.3.15 Web Browser

The users will access the application through the web-browser. The web-browser will view exported reports in HTML or XML formats.

4.3.16 E-mail Server

The application can send notification e-mails to the users through the e-mail server.

4.3.17 E-mail Reader

The application will provide many notification e-mails and reset password e-mails, so the user has to have and use his/her e-mail reading software to receive and read that information from the application.

4.3.18 Offline Data Storage

Some offline data storage (like removable media) can be used to transfer data between public and private instances of the Gaming Website.

4.3.19 Local File System

The application will upload source code, tools and games from the local file system of the user. It will be also possible to download software verification formal output (documents) to local file system.

4.3.20 PDF Reader

The user needs to have some PDF reader application to view exported reports in PDF format.

4.3.21 Text Reader

The user needs to have some text reader application to view exported reports in CSV, XML, JSON, and TXT formats.

4.3.22 Twitter

The application will share user data on Twitter social network.

4.3.23 Facebook

The application will share user data on Facebook social network.

4.3.24 CD/DVD Recorder

The application will generate LiveDVD and store it to some CD or DVD disk by using some CD/DVD recorder. CD/DVD recorder can be also use to install the private instance of Gaming Website from LiveDVD or run the private instance of Gaming Website from that LiveDVD without installation.

4.3.25 FTP Server

It will allow data transfer to the public Gaming Website from the software owner and vice versa. Scheduled reports will be stored to some FTP servers by the Gaming Website, and they can be downloaded from it by the user.

4.3.26 Private Gaming Website Instance

It will be installed or simply run from the LiveDVD. Private instance of Gaming Website will allow custom usage of the Gaming Website on the customer's site by the limited set of users (i.e. staff of that customer).

4.3.27 Logging System

The application will log all errors, exceptions, warnings and debug information to some Logging System (it can be like simply log files).

4.3.28 Human Dependencies

4.3.28.1 Anonymous User

Please refer chapter 4.1 for details.

4.3.28.2 Registered User

Please refer chapter 4.1 for details.

4.3.28.3 Game Player

Please refer chapter 4.1 for details.

4.3.28.4 Site Administrator

Please refer chapter 4.1 for details.

4.3.28.5 Game Administrator

Please refer chapter 4.1 for details.

4.3.28.6 Software Owner

Please refer chapter 4.1 for details.

4.3.28.7 Reporter

Please refer chapter 4.1 for details.

4.3.28.8 External System

Please refer chapter 4.1 for details.

4.4 Task Requirements

Task Requirements correlate to the use cases, above. There should be at least one task for each use case bubble. Each use case should be briefly described. If there are specific requirements related to the use case, they should also be listed. This section captures high-level, basic functional requirements. Detailed requirements will be defined during the Requirements Specification contests.

Please note, the following use cases are repeated multiple times below, because they have to be fully implemented not in one sub-system, but in multiple different modules:

- Track Analytics Data,
- Perform Auditing,
- Perform Logging,
- Localize GUI Screens.

“M1, General” Functionality

4.4.1 View Homepage

- The anonymous user will open the application and get the homepage with the data and hyperlinks to application features, related to that user. Please note, registered users will also see a homepage that contains data relevant to his/her user role. Administrators and Reports will get homepages with more features.
- The most important data and most popular functionality will be directly available on the user's homepage.
- There will be a navigation menu, hyperlinks to top parts of the application and service information, some news, current games, etc. data on the homepage.
- Please note, registered users will additionally have a customized dashboard – refer chapter 4.4.26.
- It dedicated to be a first page of the Gaming Website web-application, displayed when the user opens the application.
- Actual presentation of the homepage is up-to the future Studio competitions, but some basic considerations are as following:
 - Homepage will provide logo, branding and title of the Gaming Website application.
 - Ability to perform searching of the web-site data is needed.
 - Links to login/logout and register to the application are needed.
 - Links for help, FAQ, Terms & Conditions information, rules, and contacts are needed.
 - News information should be displayed.
 - Navigation to view various content (forums, blogs, wiki, posted games) is needed.
 - Ability to navigate to tournaments, contests, and team competitions is needed.

4.4.2 Get News

- The application will display the news for all the users. It is needed to always have users tuned with Gaming Website application evolution and informed about upcoming event, new features, etc.
- There will be articles about new Gaming Website features in the news.
- Announcements, notifications, special events, important awards (etc.) information can be also provided through the news.
- The news entry needs a title and rich-text formatting body. Pictures are also allowed in the news article, but they are optional.
- The news will be briefly listed (like the header and the first chapter).

- The user can choose desired news entry and view the details.

4.4.3 View Forums

- There will be various forums on Gaming Website. It will be needed for discussion of the games, competitions, software verification/evaluation and just for inter-users communication.
- The user can view forum threads and posts from the community users.
- Forum is expected to be a standalone system, integrated to the Gaming Website application.
- Some open source forum software is needed to reduce development time and get a powerful solution.
- Standard forum functionality will be present (threads, text posts, images and attachments will be supported). Non-registered users can only read the forums, but cannot post/submit anything to the forum.
- Searching of forums by keywords, user handles, dates range will be supported.
- Forums will be moderated to not allow illegal messages and improper content.

4.4.4 View Blogs

- There will be personal blogs, supported for each of the registered users, but non-registered users can view those blogs if the registered user did not restrict that in his/her profile. This feature is needed to motivate newcomers by looking into personal experience with the already present users of Gaming Website.
- The user can view blog articles and comments from the community users.
- Blog is expected to be a standalone system, integrated to the Gaming Website application.
- Some open source blog software is needed to reduce development time and get a powerful solution.
- Standard blog functionality will be present (like articles with rich formatted text and images). Non-registered users can only read the blogs, but cannot post/comment anything to the blog.
- Searching of blogs by keywords, user handles, dates range will be supported.
- Blogs will be moderated to not allow illegal messages and improper content.

4.4.5 View Wiki

- There will be wiki pages, written by community members for the Gaming Website
- The user can view wiki pages and comments from the community users.
- Wiki is expected to be a standalone system, integrated to the Gaming Website application.
- Some open source wiki software is needed to reduce development time and get a powerful solution.
- Standard wiki functionality will be present (like articles with rich formatted text and images). Non-registered users can only read the wiki pages, but cannot post/comment anything to the wiki.
- Searching of wiki pages by keywords, user handles, dates range will be supported.
- Wiki pages will be moderated to not allow illegal messages and improper content.

4.4.6 View Posted Games

- The user can view the list of posted games.
- The list will show the name of the game, available levels, schedule and the game progress.

- The list of participants will be shown for the posted games too.
- A very brief description will be provided for the posted games.

4.4.7 View Tournaments

- The user can view data about long term on-going tournaments (like multiple game competitions between multiple users) and team competitions.
- Name, rules, pre-requisites and required user skills, date/time period, virtual/physical prizes, etc. information will be shown for each tournament and team competition.
- Historical information of the tournaments will be available too.

4.4.8 View Contests

- The user can view information about currently performed short contest (like single game competitions between multiple users).
- Name, rules, pre-requisites and required user skills, schedule, virtual/physical prizes, etc. information will be shown for each contest.
- Historical information of the contests will be available too.

4.4.9 Perform Searching

- The user can perform basic searching (like by keywords) for the needed content.
- Advanced searching (like by complex search criterions) can be used too.
- The application will match the content of the Gaming Website according to the entered search criteria and display the list with the most relevant found results.
- The user can press any of the found entry and view detailed content page.

4.4.10 Get Help

- The user can freely access all the help information about the application – like by pressing “Help” hyperlink or button.
- Brief help on the application usage will be shown for the user.
- Some visually attractive help (like getting started) is needed, so the user will be quickly and easily get involved in the usage of the application.
- The user guide and troubleshooting guides can be also retrieved by the user.

4.4.11 Get FAQ

- The user can view FAQ page with information about most common questions and answers.
- The user can view the list of the most common questions and the related answers.
- Standard FAQ functionality will be supported.

4.4.12 Get Terms and Conditions

- The user can view Terms & Conditions page with legal information on the application usage.
- It will be a static page with the related legal textual information.
- The user will be required to agree with terms and conditions until registering to the Gaming Website.

4.4.13 Get Rules

- The user can view rules of Games Server and software verification approach.
- It will be a static page with the related information on the Gaming Website usage, basic game principles and common playing rules.

- The rules can be clarified by the related pictures.

4.4.14 Get Contacts

- The application will provide full contacts information about its owner and support team. It is needed to physical contact from users to the application owners and support team – like to get answers, immediate troubleshooting on application usage, reporting concerns/issues, etc.
- Mailing address, phones & faxes, e-mail address and contact persons will be listed.
- It is assumed there will be multiple contact person for various purposes (like public relations, competition managers, etc.) – they contacts will be publicly available from the application.

4.4.15 Print Web-pages

- Any web-page of the application (including leaderboards, reports, news, etc.) can be printed by the user.
- The application will prepare printer friendly version of the web-page on the user request.
- The application will use printing functionality from the user's web-browser to print any of its web-pages.
- Pagination during printing will be performed by the internal functionality of the user's web-browser.

4.4.16 Play Sample Game [external]

- The anonymous user can freely play some sample games for software verification.
- Only very simple games will be available without registration.
- The games will be played on the user's computer or on the mobile device.
- Games are provided externally – this is out of scope for this conceptualization.

4.4.17 Register to Application

- The non-registered user can register to the application. This feature is needed to allow users have a personal space on the application, be identified by handle/name during competitions and other activities, collect user achievements.
- Just one registration account will be possible to each person.
- Registration process will be very simple and the application will minimize amount of personal information collected from the user.
- Just e-mail address, username and password, date of birth (or just handle and password from some social network) will be enough for registration.
- The user will be required to enter CAPTCHA code to prevent robot/spammer registration.
- The user has to accept term and conditions to register to the application.
 - The application will display a page with terms & conditions on the user request and the user have to explicitly mark that he/she is agreed to – otherwise he/she cannot register. Please refer to the chapter 4.4.12 for more details.
- Registration will require activation through the user's e-mail.
- The application will create the user account and default dashboard after the user activated his/her registration through e-mail.
- The user will be registered as a Registered User and a Game Player, but more user roles can be added later to his/her account by the Site Administrator.
- The notification e-mail to the new user is needed after successful registration.

4.4.18 Login

- The registered user can login to the application. This feature is needed to provide access to various parts of the application functionality according to the user role and to track/control user's activity with the application.
- The user will enter his/her handle and password to login.
- Login will be performed with the help of some internally implemented Security System (like database).
- Login will be integrated with OAuth (for authorization of users) and OpenID (for authentication of users).
- The application will authenticate and authorize the user according to his/her user role (i.e. Game Player, Site Administrator, etc.) and provide the related functionality.
- In case of successful login the user will get access to the related functionality (according to user role) in the application.
- Else (if login failed), then the application will inform the user what provided credentials are wrong.

4.4.19 Logout

- The logged in user can logout from the application at any time. This feature will be used to improve security when the user has no more needs to use the application.
- The logged out user will lost access to his/her previously available functionality in the application – he/she will get just read-only functionality from the application (like non-registered user).

4.4.20 Recover Forgotten Password

- There will be a “forget password” feature (like a link) for registered users, who cannot remember the password for their accounts. It is needed to help the user restore access to his/her account after the user forgot his/her password. Otherwise (if that feature was absent) the user may try to create one more account, which is prohibited.
- The system will send the password restore e-mail to the user's e-mail address, provided during registration, on that user's request.
- The user will read that e-mail and reset the password to access his/her account.
- After that, the user will be able to login and enter a new password for his/her account.

4.4.21 View User Profiles

- There will be an ability for the registered user to quickly get access to his/her own account on the Gaming Website.
- The user can freely view his/her account and all the data entered by him/her to the account.
- It will be possible to search for and view accounts of other users that shared their data on the Gaming Website.
- It will be possible to open the user account by pressing hyperlink on the username of the related user on the Gaming Website.
- The user profile will be display like a personal page (visit card) with the data that was explicitly allowed for sharing by the profile owner.
- Only e-mail address, username (handle), password, and date of birth are required fields of the user profile.

4.4.22 *Customize Profile*

- The registered user will be able to edit his/her own account and provide more personal data (if needed).
- The application will not require the user to provide more personal data (than just e-mail, username, password, date of birth), but the user can optionally enter that data and share it with other community members.
- The user can provide text (rich formatted) about him/herself to publish in the profile.
- Usage of some CMS is needed to allow profile customization, placing free form text data to the profile and arranging the data in the profile. It will simplify user ability to customize his/her profile.

4.4.23 *Share User Data*

- The registered user can share his/her data (profile status, last posts on the forum, blog posts, wiki pages, awards, etc.) with other users and through social networks. This feature is needed to have powerful communication and social relations between users of Gaming Website application.
- The user can choose which data from his/her profile to share with other community users, and which data records from his/her profile have to be hidden.
- Sharing of users' data will be supported for Facebook and Twitter popular social networks.

4.4.24 *Unregister from Application*

- The registered user can unregister from the application if needed (like to terminate his/her account and to never more login to the application).
- The user will confirm his/her decision and the application will remove the user account, profile, and the related data.

4.4.25 *Get Notified by E-mail*

- The registered user can be notified by the application in various cases. This feature is needed to additionally inform the user about what is new and what has happen on the education web-portal.
- The user will get notification on his/her account confirmation, successful registration to the application, on new content and contests, on award announcements, etc.
- The user can choose the format of the e-mail message: HTML or plain text.
- The user can flexibly configure which types of notifications he/she would like to receive from the application and which are not.

4.4.26 *View Dashboard*

- The registered user will have an own space page on the Gaming Website. It is needed to simplify the user's experience with the application and have all the most important parts of the application easily accessible by the user.
- It will be like a dashboard, or the user's "My Space", where the most important information about the current activities from the Gaming Website, news and related data (forum posts, chat messages, announcements) will be displayed.
- The navigation links for other features of the Gaming Website application will be also displayed on the user's dashboard.
- The dashboard will simultaneously input data from various places of the application and present them on a single page.
- Displaying of various set of data will be provided through small windows – widgets.

4.4.27 *Customize Dashboard*

- The registered user can customize his/her dashboard.
- The user can perform standard actions with the dashboard widgets – like moving, re-sizing, selecting page layout.
- It will be possible to add needed widgets from the list of available ones and to remove widgets from the dashboard as needed.
- Types of data, fetched by the dashboard, will be customizable.

4.4.28 *Perform Blogging*

- The registered user can post articles to his/her personal blog and comment other users' blogs. This feature will allow the user to share his/her personal experience on Gaming Website application with other users.
- The user can enter rich text formatted articles (with images) to the blog.
- Full standard blogging functionality will be provided for registered users when working with their own blogs. They cannot post articles to blogs of other users.
- The registered user can post comments to other user's blog and answer to comments on his/her own blog as well.
- It is assumed that blog will be a standalone system – like re-using of the existing open source software. It is most likely, some CMS is needed here (like Liferay), because implementing a blog through a CMS is very easy and the solution will be very powerful.
- The blog articles will be moderated by the Site Administrator to prevent illegal and improper content.

4.4.29 *Write on Personal Walls*

- The registered user can write data on his/her own personal wall and on the walls of other community members.
- Simple text messages, free drawing (through a very simple graphical editor), and pictures are allowed for placing on the personal walls.
- The personal walls will be moderated by the Site Administrator to prevent illegal and improper content.

4.4.30 *Participate in Forums*

- The registered users will be able to view and post threads/comments to the forums. This feature is needed to achieve an efficient communication/discussion between the application users.
- The user can provide text-based posts, attach files to the forums and vote on posts from other users.
- The user can download attachments as well.
- Full standard forum functionality will be supported for registered users.
- Please note, the forum is expected to be a standalone system (like re-use of some open source forum).
- The forum posts and attachments will be moderated by the Site Administrator to prevent illegal and improper content.

4.4.31 *Prepare Wiki Pages*

- The registered user can create articles on the wiki and comment articles of other community members.
- The user can provide rich text-formatted articles and place pictures to the wiki.
- Full standard functionality for wiki pages will be supported.
- It is assumed that blog will be a standalone system – like re-using of the existing open source software.

- The wiki pages will be moderated by the Site Administrator to prevent illegal and improper content.

4.4.32 *Post through Social Networks*

- The registered user can perform data exchange with common social networks (like Facebook, Twitter).
- Standard social networks' posting will be supported.
- The user can post some game-related articles, "like" Gaming Website, re-publish data, etc. through those social networks.

4.4.33 *Perform Messaging*

- The registered user can send private messages to other users of Gaming Website application. It is needed for instant communication between all the community users.
- The user can view text messages from other community members and send own text message to chosen community members in real-time (like textual chat).
- Standard chat features will be supported and; all the incoming and outgoing messages will be displayed in the special window on the user's dashboard.

4.4.34 *View Game Showcase*

- The application will provide interesting showcases (like visually appealing presentations) on the games and registered users can freely view that data before start playing the related game.
- If the registered user is interested in some game for software verification, then he/she can view the showcase for that game.
- The application will display the visually appealing brief presentation of the chosen game.
- There will be some sample screens of the game, it's goal, brief description, detailed rules, hints and tips.

4.4.35 *View Leaderboards*

- The registered user can view leaderboards with top performers for the finished competition and sets of competitions. It is needed to motivate newcomers and possible new users to join the application.
- There will be several different leaderboards, separated by topics.
- The user can view leader boards with summary skills of community members and their overall places (top 10, 25, 50, etc.).
- There can be a single common leaderboard, and special leader boards per each contest, tournament, team competition, and game.

4.4.36 *View Score Statistics*

- The application will show the user scores and placements for the played games.
- The user can view scores and placements for other community members for the played games.
- User scores/ratings, skills coloring (like on TC web-site), and handles will be displayed, but there will be no more personal data of the users.

4.4.37 *View Awards*

- The application will display user awards and achievements (like top places, fast speed, the best optimal solution, etc.).
- The user can view awards for other community members too.

4.4.38 View Virtual Badges

- The application will display user badges – like some small pictures representing the user skills, experience, and exciting winnings.
- The user can view badges for other community members too.

4.4.39 View Prizes

- The application will display any physical prizes or gifts earned by the user for paid games, contests, tournaments (and team competitions).
- Information about prizes/gifts earned by other community members will be only viewable if those community members agreed to share that personal data.

“M1, System” Functionality

4.4.40 Host Gaming Website

- The System of the module M1 will host the public Gaming Website on some dedicated web-server, so it will be freely available to the user from the Internet.
- The Gaming Website is a regular web-application available online from regular computers and mobile devices.

4.4.41 Publish Games

- All the games will be published on the Gaming Website, so the user can load it and play either on his/her PC or on the mobile device.
- The games are developed externally (out of scope for this conceptualization). They will be uploaded and listed on Gaming Website.
- The brief description and rules of the game will be provided for each game.
- The games will be accessible by users, which can load, run, and play them.

4.4.42 Communicate with Private Instances of Gaming Website

- If the private instance of Gaming Website is installed on some customer's site, then the public Gaming Website can communicate with it.
- The System will provide needed communications between the public Gaming Website (installed on the Internet) and private instance of the Gaming Website (installed from LiveDVD).
- Game levels, game results and user solutions can exchanged between the public and private instances of the Gaming Website.

4.4.43 Integrate with Mobile Devices

- The system of the module M1 will be integrated with mobile devices when playing chosen games.
- The system will perform needed integration between the Gaming Website and native game played on the mobile device by transferring related data between web-site and the mobile device.

4.4.44 Integrate with Social Networks

- The system of the module M1 will provide sharing of user data on Facebook. It is needed to have efficient communication of the user with his/her friends and to implicitly promote Gaming Website application on the popular social network.
- Sharing data on Facebook will be under full control of the user – he/she will ask the application what to share and it will transparently deliver his/her data to Facebook.

- The system will provide sharing of user data on Twitter. It is needed to have efficient communication of the user with his/her friends and to implicitly promote Gaming Website application on the popular social network.
- Sharing data on Twitter will be under full control of the user – he/she will ask the application what to share and it will transparently deliver his/her data to Twitter.
- User profile statuses, recent awards, competition results, forum threads, blog articles are good examples of data sharing for Facebook and Twitter.

4.4.45 *Send Notification E-mails*

- The system of the module M1 will send various e-mail notifications to various uses in several events during the application execution. It is needed to keep users up-to-date with new activities, performed on the education web-portal.
- The possible cases are:
 - Confirmation of a new account,
 - Successful registration on the new account,
 - New content published,
 - New contests, tournaments, or team competitions published,
 - New awards were assigned,
 - Featured events,
 - Etc. – up-to Architect.
- The system will provide descriptive subject (like with the information about the Gaming Website application and the e-mail purpose).
- The system will provide the body text with clear and detailed explanation of the message information.
- The system will provide the brief information about the related game/contest/tournament/team competition, achieved scores, placement and type of the award in case of the user earned some achievements
- The system will send the notification e-mail to the e-mail address of the related user through some E-mail server.

4.4.46 *Provide General Web Services API*

- There will be a special General Web Services API providing programmatic access to the general functionality of the gaming web site and public community (like achievements, conversations, application general information, etc.).
- Some authentication (integrated with OAuth) will be required to access that API.
- The API will allow external sub-systems to easily integrate with general functionality of the web-site, to provide the general data to the Gaming Website and to retrieve existing general data from the Gaming Website.

4.4.47 *Track Analytics Data*

- The System of the module M1 will automatically track all the statistics and analytical data on the application usage, users and games during work of the Gaming Website.
- Game statistics data, user statistics data, and application usage information will be tracked.
- The data is tracked to the database and can be used later when creating reports for Administrators.

4.4.48 *Perform Auditing*

- All the user activities will be audited by the application. It is needed to store what actions which users do with the application – it can be used in maintenance/testing and settling issues with illegal content/behavior of users.
- The application will audit the name of the user action, like:

- Register,
 - Login/logout,
 - Post forum thread,
 - Post in blog,
 - Comment in blog,
 - Post in wiki,
 - Comment in wiki,
 - Post on personal wall,
 - Play game,
 - Participate in the contest, tournament, or team competition,
 - Etc. – all the user activity must be audited.
- The following data will be stored together with the action name:
 - Date/time stamp,
 - User handle,
 - IP address of the user (if it is possible and is not restricted for some groups of users),
 - Previous data value (if any),
 - New data value (if any).
- Data will be audited to the database – more details are up-to the Architect.
- No user credentials can be audited.

4.4.49 Perform Logging

- The system of the module M1 will log all the errors, exceptions, warning and debug information during the application execution. It is needed for further maintenance, support and improvement of the application.
- Some sort of the logging system (like just files) is needed for storing logging information.
- The logged information must not contain user credentials.
- Logging will have various levels (configurable) and can be even turned off (configurable) to improve speed of the application.

4.4.50 Localize GUI Screens

- The system of the module M1 will automatically localize all GUI screens of the module according to the user's country and language (configurable).
- All the strings and text labels of the GUI will be localizable.
- Data formats of all the fields in the GUI will be localizable.
- Different time zones will be supported for users all over the world.

"M2, Game Player" Functionality

4.4.51 Choose Game to Play

- The Game Player can select which games and levels to play from the set of the currently available software verification games.
- Some game levels can be freely chosen by him/her, others can be assigned by the Game Administrator for concrete users or users with the required skills (please refer to the use case 4.4.96).
- More advanced game levels will only available for highly skilled users. Those levels can be adaptively assigned as described in the chapter 4.4.60.

4.4.52 Run Game

- The Game Player can run the chosen game to start playing it.
- The game will be loaded from the Gaming Website and executed in the web-browser on his/her PC or the mobile device.

4.4.53 Play Game on PC [external]

- The Game Player can play software verification games, published on the Gaming Website, on his/her computer.
- The game play will allow the user to solve some task, puzzle or to find some solution in logical model. It is expected that will be directly related to the verification of the related part of the source code for the software product under evaluation.
- The game task will depend on the game type and the game level.
- Advanced game levels will have more tasks, more difficulties and complexity than initial simple game levels.
- Software verification games for PC are implemented externally, so playing them is out of scope for this conceptualization.

4.4.54 Play Mobile Game [external]

- The registered user can play software verification games, published on the Gaming Website, on his/her mobile device.
- The game can be played on the mobile device in the similar way as it was described in the chapter 4.4.53 for PC game play.
- The system of the module M2 will perform needed integration between the Gaming Website and native game played on the mobile device, so the data is properly exchanged between the mobile device and the Gaming Website website.
- Software verification games for mobile devices are implemented externally, so playing them is out of scope for this conceptualization.

4.4.55 Participate in Tournaments

- If there are some tournaments, then the Game Player can attend them on the Gaming Website. Those competitions can have some predefined prizes/gifts or virtual awards/badges.
- The Game Player can participate in long tournaments (usually involved many game levels or even multiple games) and team competitions. He/she can register to the contest and participate in the tournament-related competitions.
- The system will automatically calculate user's scores, placements and achievements earned through the tournament in addition to results of a single game play.

4.4.56 Participate in Contests

- If there are some contests, then the Game Player can attend them on the Gaming Website. Those competitions can have some predefined prizes/gifts or virtual awards/badges.
- The Game Player can attend special contests – like short competitions on a single game with multiple other game players.
- He/she can register to the contest and participate in the contest-related competitions.
- The system will automatically calculate user's scores, placements and achievements earned through the contest in addition to results of a single game play.

“M2, System” Functionality

4.4.57 *Integrate Games*

- The system of the module M2 will integrate all the games to the Gaming Website, so the user can load it and play either on his/her PC or on the mobile device.
- All the games are prepared externally (by Game Developers – out of scope) and registered to the Gaming Website through Game API.
- Some generic database schema will be used for data storage and easy integration with any games.
- Integration of game levels can be implemented through the XML files with game level maps, layers, graphics, objects, physics, and game logic – it is a standard approach, but it is up-to the Architect.
- All the temporary data of the game are persisted to the database.
- The System will store all the user achievements (including scores, placements, awards, badges, physical prizes/gifts) to the database.
- The System will acquire all the game results and user optimal solutions.
- The game results and optimal solutions found by the users will be stored to the database.
- Please refer to the use cases 4.4.63 - 4.4.74 for more details on game integration.

4.4.58 *Integrate Adaptive Scheduling*

- The system of the module M2 integrates special abilities of smart scheduling of game levels for playing by users with enough skills.
- Adaptive scheduling itself is out of scope for this conceptualization, but all the infrastructure for its future integration will be provided.
- The following integration points are needed:
 - Calculation of users' skills,
 - Automated assignment of game levels to users according to their skills,
 - Scheduling games for users of appropriate skills,
 - Scaling of a single game through multiple users.
- Please refer to the use cases 4.4.59 - 4.4.62 for more details on the expected adaptive scheduling to be implemented in the future.

4.4.59 *Determine User Skills [external]*

- In the future, the System of the module M2 will integrate calculation of users' skills with adaptive scheduling process. Please note, that currently this use case is external (out of scope for this conceptualization). A single set of skills will be determined individually per a single user.
- The user's skills level will be determined according to his/her scores, count of played games, difficulties of finished game levels, time used for solving problems, count and quality of found solutions, etc.
- The Architect has to develop several most important categories of user's skills (like scores, speed, quality, etc.) and design calculation formulas for them.
- Some sort of aggregation formula is also needed to determine the overall skills of the game players.
- The aggregate user's skills will be used to properly assign game levels and schedule games (please refer to use cases 4.4.60 and 4.4.61).

4.4.60 *Assign Game Levels (adaptively) [external]*

- In the future, the system of the module M2 will determine which game levels can be allowed for playing by the related user according to his/her skills. Please note, that currently this use case is external (out of scope for this conceptualization).

- Matching of advanced game levels to the related game players will be performed according to the skills/performance/experience of those game players, so only highly skilled players can advance to more complicated game levels.
- There isn't full flexibility on game levels selection for Gaming Website users. Users can manually choose just several initial levels, but they have to perfectly play many games and get enough skills to proceed to more advanced game levels.
- The system will automatically assign advanced game levels to the matched game players.
- Please note, advanced and important game levels can't be manually chosen by unskilled new game players.

4.4.61 *Schedule Games (automatically) [external]*

- In the future, the system of the module M2 can automatically schedule games for a set of users – like during special contests, tournaments, or team competitions. Please note, that currently this use case is external (out of scope for this conceptualization).
- The system will automatically schedule games according to the count of enough active game players, skills/experience/performance of game players, and software products posted for verification and evaluation.
- If any new software product becomes available, then the system will automatically schedule new contests, tournaments, and team competitions for it.
- The system will automatically schedule contests, tournaments, and team competitions according to count of software products available for verification on the Gaming Website and count of currently active game players.

4.4.62 *Scale Game Play [external]*

- In the future, the system of the module M2 will highly scale playing a single game or game level through many Game Players. Please note, that currently this use case is external (out of scope for this conceptualization).
- Multiple Game Players will be able to play the same game in parallel to ensure that at least some of them will successfully finish and produce quite appropriate results.

4.4.63 *Provide Levels Data to Games*

- If the game was started by the user, then the system of the module M2 will retrieve the generated and selected game level data and use it during game play.
- The system of the module M2 will provide seamless and transparent access to game level data from the storage to the user during game play.
- It is suggested to use XML files with game level maps, layers, graphics, objects, physics, and game logic – it is a standard approach (like in Tiled game level editor: <http://www.mapeditor.org/>), but it is up-to the Architect.

4.4.64 *Retrieve Game Instance for User*

- If the game was started by the user, then the system of the module M2 will retrieve the game instance and provide it for running on the user's PC or on the user's mobile device.
- The system of the module M2 will provide seamless and transparent delivery of the game from the storage to the user.
- It is suggested to use a single archive file for storing/retrieving a set of game program files, resources, meta data, and other information – up-to the Architect

4.4.65 Persist Game Play

- The System of the module M2 will persist all the data during game play.
- All the temporary data of the game, user results and solutions are seamlessly and transparently persisted to the database.
- It will be possible to retrieve all the persisted data through the Game API.
- The database schema for that data is up-to the Architect.

4.4.66 Store User Achievements

- The System of the module M2 will store all the user achievements (including scores, placements, awards, badges, physical prizes/gifts) to the database.
- The Gaming Website will seamlessly and transparently persist all the calculated metrics of the user achievements.
- That data can be used later for displaying to the user.
- It will be possible to retrieve all the persisted data through the Game API.
- The database schema for that data is up-to the Architect.

4.4.67 Acquire Game Results and Solutions

- The System of the module M2 will acquire all the game results and user optimal solutions.
- The Gaming Website will transparently and seamlessly get that data from the played games.
- The game results have to be dynamically transferred between the game and Gaming Website during game play.

4.4.68 Store Game Results and Solutions

- The game results and optimal solutions found by the users will be stored to the database.
- The Gaming Website will transparently and seamlessly store user results and optimal solutions from games to the database.
- It will be possible to retrieve all the persisted data through the Game API.
- The database schema for that data is up-to the Architect.

4.4.69 Determine Game Completion [external]

- The Game will automatically check if the game level was completed – it will be performed by some algorithm in the level.
- Special game level completion algorithms are up-to the Game Developers (out of scope).

4.4.70 Store Game Completion

- If the game is completed (i.e. enough results and user solutions are collected), then that information will be stored to by the system.
- The game or game level completion status will be persisted to the database.
- It will be possible to retrieve all the persisted data through the Game API.
- The database schema for that data is up-to the Architect.

4.4.71 Calculate Users Achievements

- The user can be awarded by some achievements if he/she performed well and got high scores.
- The system of the module M2 will assign related awards, badges, winning places, physical prizes/gifts to the top placed users for the played games.

- The system will calculate the users' ratings and the related colors (like on the TC web-site for competing members).
- The calculation formula for achievements is up-to the Architect.

4.4.72 *Persist Verification Results*

- The Game API will provide the external verification tools an access to all the persisted game-related data.
- The verification tools can freely retrieve game results and user solutions for generating the verification results.
- The verification results can be stored to the Gaming Website's database through the web-services API.

4.4.73 *Determine Game Scores*

- The user is scored after the game completion.
- The system of the module M2 will calculate user scores for the played games according to amount, quality and speed of achieved results.
- The system will calculate user placement (in comparison with other competing users) according to his/her scores.
- Evaluation of game results will be performed by the game infrastructure and is out of scope for this conceptualization.

4.4.74 *Provide Game Web Services API*

- There will be a special Game Web Services API providing programmatic access to all the gaming functionality. Please note, all the use cases 4.4.58 - 4.4.73 are exposed as Game API.
- It will integrate games into the Gaming Website web site.
- The API will include games and users' data transmission between games and the Gaming Website.
- Game levels and results will be transmitted as well through that API.
- Some authentication (integrated with OAuth) will be required to access that API.
- The API will allow external sub-systems to easily integrate with gaming functionality of the web-site, to provide the games and game data to the Gaming Website, control games, and to retrieve existing game data from the Gaming Website.

4.4.75 *Track Analytics Data*

- The system of the module M2 will track analytics data like described in the use case 4.4.47.
- This use case is provided to clarify that analytics tracking has to be implemented not only in the module M1, but in the module M2 too.

4.4.76 *Perform Auditing*

- The system of the module M2 will perform auditing like described in the use case 4.4.48.
- This use case is provided to clarify that data auditing has to be implemented not only in the module M1, but in the module M2 too.

4.4.77 *Perform Logging*

- The system of the module M2 will perform logging like described in the use case 4.4.49.

- This use case is provided to clarify that logging functionality has to be implemented not only in the module M1, but in the module M2 too.

4.4.78 *Localize GUI Screens*

- The system of the module M2 will perform localization of GUI screens like described in the use case 4.4.50.
- This use case is provided to clarify that localization functionality has to be implemented not only in the module M1, but in the module M2 too.

“M3, Site Admin” Functionality

4.4.79 *Post News*

- If any news data is occurred or some upcoming news data is expected, then the Site Administrator can post a news entry to the Gaming Website.
- The Site Administrator will provide a regular news article (with header, rich-formatting text and, optionally, pictures) and it will be posted on the Gaming Website.
- Some sort of CMS (like Liferay, Drupal, Joomla, Magnolia) is needed to easily create/publish new posts and to hide old news articles as needed.

4.4.80 *Manage Content*

- The Site Administrator can freely manage any content on the Gaming Website.
- He/she can create new blogs, forum posts, wiki pages, write on personal walls of other users.
- He/she can edit any existing content entries of any users.
- He/she can remove any existing content entries of any users.
- All data of the content entries can be managed.
- Some sort of CMS (like Liferay, Drupal, Joomla, Magnolia) is needed to easily manage content entries on the Gaming Website.

4.4.81 *Moderate Forums*

- The Site Administrator can moderate all threads on the forums. It is needed, because the application must not provide illegal or improper content.
- Site Administrators will have a queue of all the new messages on all the forums, they will read them and can edit/remove bad messages.
- The regular functionality of the forum moderation will be re-used (from some open source software).

4.4.82 *Moderate Blogs*

- The Site Administrator can moderate all posts on the blogs. It is needed, because the application must not provide illegal or improper content.
- Site Administrators will have a queue of all the new articles on all the blogs and all new comments on the blogs.
- They will read them and can edit/remove bad blog articles and comments.

4.4.83 *Moderate Wiki*

- The Site Administrator can moderate all articles on the wiki. It is needed, because the application must not provide illegal or improper content.
- Site Administrators will have a queue of all the new articles on the wiki and all new comments on the wiki articles.
- They will read them and can edit/remove bad wiki articles and comments.

4.4.84 Moderate Personal Walls

- The Site Administrator can moderate all posts on all the personal walls. It is needed, because the application must not provide illegal or improper content.
- Site Administrators will have a queue of all the new posts on the personal walls.
- They will read them and can edit/remove bad posts from personal walls.

4.4.85 Send Announcements to Users

- If any special event has occurred, then the Site Administrator can directly announce it to chosen users.
- The Site Administrator can send special announcements to the chosen users by e-mail.
- The e-mail message will contain the information about the special event as free form text, provided by the Site Administrator.
- The e-mail message will be sent by some e-mail server – like described on the chapter 4.4.45.

4.4.86 Manage User Accounts

- The Site Administrator can freely manage all the user accounts on the Gaming Website.
- He/she can add new users, modify data of the existing users, disable/enable or even remove existing users from the application.
- Removal of the user's account will require confirmation from the site administrator and it will not be possible to remove the last administrator account from the Gaming Website.

4.4.87 Manage User Access Rights

- The Site Administrator can freely manage access rights of all the users of the Gaming Website.
- He/she can modify access rights of users and assign them more user roles (or remove some roles from users).
- It will not be possible to remove administrator role from the last site administrator account from the Gaming Website.

4.4.88 Configure Application

- The Site Administrator can freely configure the Gaming Website application.
- Configuration will be performed on some GUI pages.
- The Site Administrator can configure logging levels, notifications, credentials to e-mail server, paths for FTP server folders, etc. – up-to the Architect.

“M3, Game Admin” Functionality

4.4.89 Provide Compilers

- The Game Administrator can upload some compilers (needed for preparing games) to the Gaming Website.
- Compilers have to be integrated with the Gaming Website and they will be used for game levels creation.
- Compilers are externally provided for the Gaming Website.
- Compilers will be like small programs to be executed by the Gaming Website.
- Compilers will allow to specially prepare games for usage on the Gaming Website.

- Implementing of those compilers is out of scope for this conceptualization.
- All those compilers will be seamlessly integrated by the system.

4.4.90 *Provide Conversion Tools*

- The Game Administrator can provide external special tools for converting software product's source code to the related game levels.
- Those tools will be uploaded to the Gaming Website.
- Conversion tools will allow to generate interesting game levels from the source code of the software product to be verified.
- Implementing of those tools is out of scope for this conceptualization.
- All those tools will be seamlessly integrated by the system.

4.4.91 *Provide Verification Tools*

- The Game Administrator can provide the verification tools for conversion from achieved game results and user solutions for formal software verification output.
- Those tools will be uploaded to the Gaming Website.
- Verification tool will prepare verification reports from the game results and user solutions.
- Implementing of those tools is out of scope for this conceptualization.
- All those tools will be seamlessly integrated by the system.

4.4.92 *Provide Games*

- The Game Administrator will provide all the needed software verification games for the related software products.
- The Game Administrator will simply upload them to the Gaming Website.
- Games will perform actual verification of the software products with an interesting and visually appealing game play.
- Implementing of those games is out of scope for this conceptualization.
- The games will be seamlessly integrated by the system.

4.4.93 *Upload Software Program*

- The Game Administrator can upload the source code to the Gaming Website.
- It will be performed on some GUI page of the application.
- Some progress indicator is needed during uploading to inform the user.

4.4.94 *Select Level Generators*

- The Game Administrator will select proper tools for generating needed counts of levels for each game for software product verification and generate the needed game levels.
- Please note different tools can be used for generating different levels even for the same game.

4.4.95 *Generate Game Levels [external]*

- The Game Administrator will generate needed game levels for the games of the software product under verification.
- The Administrator will generate needed game levels for the source code by using the chosen game level generating tools – they are implemented externally and level generation is out of scope for this conceptualization.
- Some progress indicator is needed to inform the user about ongoing process.

4.4.96 Assign Game Levels to Users

- The Game Administrator can manually assign game levels to the needed users or to groups of users having required skills.
- He/she will view the list of game levels, the list of users and their skills, so the administrator can use that information when assigning game levels.

4.4.97 Select Needed Tools

- The Game Administrator will select which tools to use for the given game for verification of the related software product.
- The list of all the available tools will be shown and the user will choose which ones to use for the related software product.

4.4.98 Configure Game Options [external]

- The Game Administrator can freely configure options of the uploaded games on the related GUI pages.
- Games will be configured on their GUI pages – externally developed together with games (they are out of scope for this conceptualization).

4.4.99 Map Game Results to Verification Output [external]

- If the game was finished, then the Game Administrator can customize automated converting game results to the formal verification output for the related software product.
- The Game Administrator can use some of the documenting tool to map collected user results and solutions to the formal verification output for the verified/evaluation software product.
- It will be possible to fully customize output verification documents – like choosing needed data parts, columns, formats, etc.
- The tool will be used externally and generate all the needed formal documents on the software verification.
- The generated formal documents are stored to the database.
- The database schema for that data is up-to the Architect.

“M3, Software Owner” Functionality

4.4.100 Upload Software Program for Verification

- The software product to be verified and evaluated is provided by the Software Owner.
- The Software Owner can deliver the source code of the software product to the Gaming Website through some API (like Web Services).
- The Software Owner can deliver the source code of the software product to the system by uploading that source code to some dedicated FTP server.
- The Software Owner can provide source code of the software product directly to the Gaming Website through some GUI page. So, he/she will directly upload the source code to the Gaming Website.

4.4.101 Deliver Software Program in Offline [external]

- The Software Owner can manually provide source code of the software product to the Game Administrator on some data storage media.
- It means providing data in the offline mode – it will be performed externally for this application.

4.4.102 Convert Software Program to Game Levels [external]

- The Software Owner can convert uploaded source code to the game levels by him/herself, if he/she needs very high security and does not want to expose source code even for Game Administrators
- The Software Owner can use some available conversion tools (externally developed, out of scope in this conceptualization) to convert source code to the game levels.
- Generated game levels will be stored in the system and used for further verification of the software product.

4.4.103 Download Software Verification Results

- The Software Owner can get software verification documents from the Gaming Website.
- He/she can download the generated formal verification documents through some API or from the FTP server.
- He/she can download the generated formal verification documents on the related GUI page of the Gaming Website.

4.4.104 Receive Software Verification Results in Offline [external]

- The Software Owner can manually get the generated formal verification documents on the offline data storage (like by asking Game Administrators to manually deliver that data).
- It means retrieving software formal verification output in the offline mode – it will be performed externally for this application.

“M3, System” Functionality

4.4.105 Integrate Game Tools

- The System of the module M3 will fully integrate with externally provided game tools – compilers, game level conversion tools, and verification tools.
- All the externally provided tools will be persisted in the system database (the database schema is up-to the Architect).
- The provided tools will be published on the web-site and available for usage by Game Administrators and Software Owners.

4.4.106 Provide Admin Tools Web Services API

- There will be a special Admin Tools Web Services API providing programmatic access to all the administration functionality.
- It will integrate new tools to the Gaming Website, to manage the content, conversation, users, game tools, games, and software programs.
- Some authentication (integrated with OAuth) will be required to access that API.
- The API will allow external sub-systems to easily integrate with administration functionality of the web-site, to provide the administrative data to the Gaming Website, manage Gaming Websites and users, and to retrieve existing administrative data from the Gaming Website.

4.4.107 Track Analytics Data

- The system of the module M3 will track analytics data like described in the use case 4.4.47.
- This use case is provided to clarify that analytics tracking has to be implemented not only in the module M1, but in the module M3 too.

4.4.108 Perform Auditing

- The system of the module M3 will perform auditing like described in the use case 4.4.48.
- This use case is provided to clarify that data auditing has to be implemented not only in the module M1, but in the module M3 too.

4.4.109 Perform Logging

- The system of the module M3 will perform logging like described in the use case 4.4.49.
- This use case is provided to clarify that logging functionality has to be implemented not only in the module M1, but in the module M3 too.

4.4.110 Localize GUI Screens

- The system of the module M3 will perform localization of GUI screens like described in the use case 4.4.50.
- This use case is provided to clarify that localization functionality has to be implemented not only in the module M1, but in the module M3 too.

“M4, Reporter” Functionality

4.4.111 Get Software Verification Report

- The Reporter can get a verification report for the chosen evaluated software product.
- The Software Verification Report is a finalized formal output report on the verification of the software source code.
- The report is automatically generated according to the game results and user solutions found during playing games for the given software product on the Gaming Website.
- It will be a professionally formatted report with detailed and summary information about quality and found bugs of the evaluated software product.
- The Reporter can get verification report in one of the following formats: HTML, XML, CSV, and JSON.
- The application will display the report on the screen.

4.4.112 Get Application Usage Report

- The Reporter can get report on how frequently and widely the application is used.
- The usage of the application by the users, most popular pages and content will be reported.
- The usage of the application by the users, most popular pages and content will be reported.
- The report will be provided in the following formats: HTML, XML, CSV, and JSON.
- The data will be professionally formatted, tables and graphs will be used where applicable.
- It will be possible to filter report data by date/time range.
- The application will display the report on the screen.

4.4.113 Get Game Statistics Report

- The Reporter can get report on played games statistics and analytical information.
- The report on game results, scores, found optimal solutions will be provided.

- The report will be provided in the following formats: HTML, XML, CSV, and JSON.
- The data will be professionally formatted, tables and graphs will be used where applicable.
- It will be possible to filter report data by date/time range and by the game name.
- The application will display the report on the screen.

4.4.114 Get User Statistics Report

- The Reporter can get report on the user statistics and analytical information.
- The user statistics, preferences, and achievements report will be provided.
- The report will be provided in the following formats: HTML, XML, CSV, and JSON.
- The data will be professionally formatted, tables and graphs will be used where applicable.
- It will be possible to filter report data by date/time range and by the username.
- The application will display the report on the screen.

4.4.115 Get Customized Reports

- The Reporter can customize all the available reports.
- The Reporter can choose needed parts of the report, needed columns, data sorting, filtering, etc.
- The customized report will be displayed on the screen in one of the supported formats (HTML, XML, CSV, and JSON).

4.4.116 Get Scheduled Reports

- The Reporter can ask the application to regularly generate reports by the customizable schedule and store them to some FTP server.
- The Reporter can choose and configure the needed schedule for generating chosen reports.
- The application will automatically generate selected reports (in HTML, XML, CSV, JSON, PDF, or TXT format) by the schedule, and persist them in the system.
- The Reporter will be able to download the scheduled report from the FTP server.

4.4.117 Get Real-time Reports

- There will be real-time reports on all the data of the application.
- Statistics, analytics, and verification output results will be dynamically displayed on a special screen.
- The Reporter can view “live” charts and tables of details and summary of the Gaming Website usage, user activity, and software verification progress.
- The newest data will be automatically and dynamically updated on the screen.
- Real-time reports will be displayed in HTML format.

4.4.118 Export Reports

- The Reporter can freely export reports to the downloadable file.
- The application will allow to export any of the displayed reports in one of the following formats: HTML, XML, CSV, JSON, PDF and TXT.
- The report will be exported to a downloadable file, so the user can download it by his/her web-browser to the local file system.

4.4.119 Print Reports

- The Reporter can freely print reports.
- Any displayed report can be freely printed by the Reporter.
- The application will print the current web-page of the report by user's web-browser.
- It will be possible to print a chosen table on multiple pages, so full large reports can be easily printed.

"M4, System" Functionality

4.4.120 Transform Game Results to Formal Verification Output [external]

- The System of the module M4 will automatically convert the collected game results and found optimal solutions to the formal verification output by using the related external tool (uploaded and chosen by the Administrator).
- That transformation will be performed by the external select tool, which is out of scope for this conceptualization.
- Various verification tools will be externally implemented and integrated to the Gaming Website.

4.4.121 Store Software Formal Verification Output

- The System of the module M4 seamlessly and transparently stores the formal verification documents (generated by the external tool) to the database.
- Those documents can be later retrieved by the Reporter.
- All the stored verification results can be retrieved through the Web-services API.
- The database schema for that data is up-to the Architect.

4.4.122 Send Scheduled Reports to FTP

- The system will automatically sent chosen scheduled reports to the selected FTP servers – they can be accessed later by other users.
- All the chosen reports are automatically generated according to the configurable schedule.
- The reports are send as files in one of the following formats: HTML, XML, CSV, JSON, PDF, or TXT.

4.4.123 Provide Reporting Web Services API

- There will be a special Reporting Web Services API providing programmatic access to all the reporting functionality.
- It will allow to programmatically access to all the reports from the Gaming Website.
- Some authentication (integrated with OAuth) will be required to access that API.
- The API will allow external sub-systems to easily integrate with reporting functionality of the web-site, to control reports, and to retrieve existing verification data and reports from the Gaming Website.

4.4.124 Track Analytics Data

- The system of the module M4 will track analytics data like described in the use case 4.4.47.
- This use case is provided to clarify that analytics tracking has to be implemented not only in the module M1, but in the module M4 too.

4.4.125 Perform Auditing

- The system of the module M4 will perform auditing like described in the use case 4.4.48.
- This use case is provided to clarify that data auditing has to be implemented not only in the module M1, but in the module M4 too.

4.4.126 Perform Logging

- The system of the module M4 will perform logging like described in the use case 4.4.49.
- This use case is provided to clarify that logging functionality has to be implemented not only in the module M1, but in the module M4 too.

4.4.127 Localize GUI Screens

- The system of the module M4 will perform localization of GUI screens like described in the use case 4.4.50.
- This use case is provided to clarify that localization functionality has to be implemented not only in the module M1, but in the module M4 too.

“M5, LiveDVD” Functionality

4.4.128 Use Private Instance of Gaming Website

- All the actions of regular Gaming Website (like described in modules M1 – M4) can be performed on the privately used instance of Gaming Website by the registered user.
- The private instance of Gaming Website is for use on a customer site with access restrictions.

4.4.129 Provide Operating System Distribution

- If the Site Administrator needs to use some new distribution of the OS, then he/she can upload it to the system.
- All the uploaded OS distributions are kept and can be used later for generating of LiveDVD.
- It is expected, that Linux OS will be used for the LiveDVD and Gaming Website.
- Some progress indicator is needed during uploading to inform the user.

4.4.130 Provide Gaming Website Distribution

- If the Site Administrator needs to use some new distribution of the Gaming Website, then he/she can upload it to the system.
- All the uploaded Gaming Website distributions are kept and can be used later for generating of LiveDVD.
- The web-server distribution is needed for the Gaming Website for proper execution.
- Some progress indicator is needed during uploading to inform the user.

4.4.131 Provide Tools Distributions

- If the Site Administrator has some new verification and conversion external tools (and compilers) then he/she can upload them to the system for further distribution on LiveDVD.
- All the uploaded external tools are kept and can be used later for generating of LiveDVD.

- The tools for generating game levels and tools for converting game results and user found solutions to software formal verification output can be uploaded to the system and persisted to the database for re-using in the future.

4.4.132 Provide Games Distributions

- If the Site Administrator has some new games, then he/she can upload them to the system for further distribution on LiveDVD.
- All the uploaded games are kept and can be used later for generating of LiveDVD.
- Games can be implemented for usage on the PCs or mobile devices of the users.
- Games will be persisted to the database for re-using in the future.

4.4.133 Provide User Guide

- If the Site Administrator needs to use some new user guides (documents), then he/she can upload them to the system for further distribution on LiveDVD.
- All the uploaded documents are kept and can be used later for generating of LiveDVD.
- The user guides will be used for teaching how to use the Gaming Website from LiveDVD.
- User Guides will be persisted to the database for re-using in the future.

4.4.134 Provide Troubleshooting Guides

- If the Site Administrator needs to use some new troubleshooting guides (documents), then he/she can upload them to the system for further distribution on LiveDVD.
- All the uploaded documents are kept and can be used later for generating of LiveDVD.
- Troubleshooting guides will be used for teaching how to install and solve possible issues of LiveDVD.
- Troubleshooting guides will be persisted to the database for re-using in the future.

4.4.135 Generate LiveDVD

- The Site Administrator can select any needed OS distribution from the set of already uploaded ones for further generating of the LiveDVD.
- The Site Administrator can select any needed Gaming Website distribution from the set of already uploaded ones for further generating of the LiveDVD.
- The Site Administrator can select any needed external tool distributions from the set of already uploaded ones for further generating of the LiveDVD.
- The Site Administrator can select any needed game distributions from the set of already uploaded ones for further generating of the LiveDVD.
- The Site Administrator can select any needed user guide distribution and user troubleshooting guide distributions from the set of already uploaded ones for further generating of the LiveDVD.
- The Site Administrator will automatically generate the LiveDVD with the chosen OS distribution, Gaming Website distribution, external tool distributions, game distributions, user guide and troubleshooting guides distributions. It will be built by some backend scripts – please refer to the use case 4.4.138.
- Some progress indicator is needed during generating to inform the user.
- The LiveDVD will be a bootable CD or DVD disk (depending on the distribution size), which can be used for full installation of the Gaming Website of the customer's site.

- The LiveDVD can be also used to boot and immediately run the Gaming Website instance right from that CD or DVD disk.

4.4.136 Transfer Data Between Public and Private Gaming Websites [external]

- The data between public and private instances of the Gaming Website can be transferred in offline mode (using storage media).
- Software products, game levels, formal verification results, etc.) can be freely interchanged between the public Gaming Website and the private Gaming Website according to the needs of private Gaming Website instance owners.
- It is expected that site administrators of public and private Gaming Website will exchange data on some data storages (like removable media) and then upload/download the data through GUI of their instances of the Gaming Website.

4.4.137 Install Gaming Website from LiveDVD

- The Administrator can use prepared LiveCD for running or/and installation of the Gaming Website.
- The private instance can be installed from the LiveCD to the customer's computer and then it will be used as a private instance of the Gaming Website.
- It will be also possible to simply run the private instance of the Gaming Website directly from the LiveCD (without installation).

"M5, System" Functionality

4.4.138 Build LiveDVD Image

- There can be special backend scripts for building an image of LiveDVD.
- The image will be built on the server-side according to the chosen OS, games, tools, documentation.
- The prepared image can be downloaded by the Site Administrator.

4.4.139 Interchange Data with Public Gaming Website

- The data between public and private instances of the Gaming Website can be transferred online through some API, or by some FTP server.
- Software products, game levels, formal verification results, etc.) can be freely interchanged between the public Gaming Website and the private Gaming Website according to the needs of private Gaming Website instance owners.

4.4.140 Track Analytics Data

- The system of the module M5 will track analytics data like described in the use case 4.4.47.
- This use case is provided to clarify that analytics tracking has to be implemented not only in the module M1, but in the module M5 too.

4.4.141 Perform Auditing

- The system of the module M5 will perform auditing like described in the use case 4.4.48.
- This use case is provided to clarify that data auditing has to be implemented not only in the module M1, but in the module M5 too.

4.4.142 Perform Logging

- The system of the module M5 will perform logging like described in the use case 4.4.49.

- This use case is provided to clarify that logging functionality has to be implemented not only in the module M1, but in the module M5 too.

4.4.143 Localize GUI Screens

- The system of the module M5 will perform localization of GUI screens like described in the use case 4.4.50.
- This use case is provided to clarify that localization functionality has to be implemented not only in the module M1, but in the module M5 too.

“External System” Functionality

4.4.144 Authenticate by Web Services API

- The external sub-system (like a conversion tool, game, etc.) can get programmatic access to the functionality of the application after proper authentication.
- Web Service API will be accessible through OAuth authentication.
- The external sub-system will send the related authentication request to the Web Service and get authenticated through OAuth protocol.
- Successfully authenticated external sub-system will get access to the web-services API.

4.4.145 Access General Web Services API

- The external sub-system can get access to all the general functionality of the gaming web site through the General Web Services API.
- That API provides access to all the content and public community (like achievements, conversations, application general information, etc.) of the Gaming Website.
- It will be possible to store and retrieve the related data through the web-services API.

4.4.146 Access Game Web Services API

- The external sub-system can get access to all the gaming functionality through the Game Web Services API. That API will be used to integrate games into the Gaming Website web-site.
- The API will include games and users' data transmission between games and the Gaming Website.
- Game levels and results will be transmitted as well through that API.
- It will be possible to store and retrieve the related data through the web-services API.

4.4.147 Access Admin Tools Web Services API

- The external sub-system can get access to all the administration functionality through the Admin Tools Web Services API.
- That API will allow integration of new tools into the Gaming Website, to manage all the content, conversation, users, game tools, games, and software programs.
- It will be possible to store and retrieve the related data through the web-services API.

4.4.148 Access Reporting Web Services API

- The external sub-system can get access to all the reports, “live” statistics, and analytics data of the software verification process and Game Service usage through the Reporting Web Services API.

- It will be possible to programmatically customize reports, get scheduled reports, get exported report files.
- It will be possible to store and retrieve the related data through the web-services API.

4.5 Security Requirements

This section documents, at a high level, the basic security requirements of the application. For example, does the system require the user to log in? Should the user's identity be authenticated on just this system, or against a central authority? Are there any special or unusual security requirements, like fingerprint scanning?

4.5.1 The system will motivate all the users to register login with a unique identity, because the functionality of the anonymous users will be greatly reduced

4.5.2 Non-registered users will have read-only access to the application and can't save game progress and results

4.5.3 There was no central authority of accounts, so the user will create his/her accounts through the application. Some Site Administrator accounts can be predefined. Other users will register as Game Players, but the Administrator can modify their user roles later.

4.5.4 The system will implement an own authentication/authorization mechanism (integrated with OpenID and OAuth).

4.5.5 Access to web services API will require OAuth authorization

4.5.6 User personal information has to be properly secured. The Architect needs to select a proper solution. The application will minimize amount of the needed personal information from the users. Only user IDs will be exposed to games.

4.5.7 All the content, provided by the users, will be moderated to eliminate illegal and improper content

4.5.8 No source code will be open for game players and it will not be possible to reconstruct the software product source code from the game levels. It will significantly improve security of the Gaming Website and this is one of the key points for the Gaming Website's approach to software verification.

4.5.9 Usage of SSL and HTTPS connection is required for this application.

4.5.10 Some private instances can be hidden by firewall or not connected to the Internet, so data transfer from them to the public Gaming Website is not possible in online. Usage of some removable media is possible for offline data transfer in this case.

4.6 Performance Requirements

Performance Requirements for the application are defined at a high level below. If there are specific requirements for specific features to perform at a quantifiable level, they too are listed below. These requirements will be developed in more detail when the Requirements Specification is written, in a later phase.

4.6.1 The expected maximum count of unique users is tens or hundreds of thousands.

4.6.2 The expected maximum count of concurrent users is thousands

4.6.3 The amount of source code to be verified can be large – like the Linux kernel

4.6.4 The application will load any web-page in 2 seconds (or less). The application has to be super fast, because performance is a critical objective for this project.

4.6.5 The application will be available 24x7 and achieve 99.9% uptime

4.7 Data Migration

Data Migration describes the data that needs to be moved from an older or external system to the new system, in order for it to operate at launch. Migration also includes data that must be transferred from the new system to another external system. Any data migration requirements are listed below.

4.7.1 *No Data Migration*

This is a fully new application, so there will be no migration of the old data to the new system.