WPF aplikacija (Model klijent-server)

Igraonica

Verzija 1.0

Istorija revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 05.06.2021. | 1.0 | <details> | Danijela Milanović, Stefan Jokić |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Positioning 3

2.1 Business Opportunity 3

2.2 Problem Statement 3

2.3 Product Position Statement 3

3. Stakeholder and User Descriptions 3

3.1 Market Demographics 3

3.2 Stakeholder Summary 3

3.3 User Summary 3

3.4 User Environment 3

3.5 Stakeholder Profiles 3

3.5.1 <Stakeholder Name> 3

3.6 User Profiles 3

3.6.1 <User Name> 3

3.7 Key Stakeholder or User Needs 3

3.8 Alternatives and Competition 3

3.8.1 <aCompetitor> 3

3.8.2 <anotherCompetitor> 3

4. Product Overview 3

4.1 Product Perspective 3

4.2 Summary of Capabilities 3

4.3 Assumptions and Dependencies 3

4.4 Cost and Pricing 3

4.5 Licensing and Installation 3

5. Product Features 3

5.1 <aFeature> 3

5.2 <anotherFeature> 3

6. Constraints 3

7. Quality Ranges 3

8. Precedence and Priority 3

9. Other Product Requirements 3

9.1 Applicable Standards 3

9.2 System Requirements 3

9.3 Performance Requirements 3

9.4 Environmental Requirements 3

10. Documentation Requirements 3

10.1 User Manual 3

10.2 Online Help 3

10.3 Installation Guides, Configuration, and Read Me File 3

10.4 Labeling and Packaging 3

A Feature Attributes 3

A.1 Status 3

A.2 Benefit 3

A.3 Effort 3

A.4 Risk 3

A.5 Stability 3

A.6 Target Release 3

A.7 Assigned To 3

A.8 Reason 3

Detalji aplikacije

# Cilj dokumenta

Dokument Detalji aplikacije govori o informacionom sistemu igraonice i pristupu bazi podataka . Aplikacija je izrađena po modelu klijent-server.   
Sistem omogućava kompletno upravljanje svim podacima u sistemu sa različitom mogućnošću što zavisi od privilegija korisnika. Pristup aplikaciji u jednom trenutku može imati samo jedan korisnik kako bi se izbjeglo konkurentno korištenje.   
Ovakav pristup je obezbjeđen na 2 nivoa. Pristup je obezbjeđen u vidu administratora i zaposlenika igraonice. Administrator ima uvid u zaposlene i računare kao i dozvolu modifikacije starih podataka i upisivanja novih. Zaposlenik igraonice ima uvid u zauzeće računara, kao i spisak igrica koje igraonica posjeduje, i na kom računaru su one instalirane.   
Sistem donosi novinu u pogledu pristupa podacima, kao i mnogo brži način pristupačnosti i nove mogućnosti koje aplikacija posjeduje.

# OPIS MODELA KLIJENT-SERVERA INFORMACIONOM

## Postavke problema

[Provide a statement summarizing the problem being solved by this project. The following format may be used:]

|  |  |
| --- | --- |
| *Problem je* | usaglašenost administrator sistema i ostalih zaposlenika igraonice, složenija instalacija sistema, |
| *Pogađa* | Administratora sistema i ostale zaposlene u igraonici. |
| *Posledice su* | dobijanje lošijih rezultata produktivnosti od očekivanih. |
| *Uspješno rješenje će* | pospješiti produktivnost i olakšati rad zaposlenicima igraonice. |

## Postavka pozicije proizvoda

|  |  |
| --- | --- |
| *Za* | Igraonicu (kao privrednog subjekta, jednu ili više) i zaposlenike te igraonice |
| *Koji* | Vode evidenciju o svojim računima prodatih usluga, računarima, klijentima i zaposlenicima. |
| *Proizvod je* | je informacioni sistem koji je prezentovan u vidu Desktop Windows aplikacije |
| *koji* | omogućava olakšano vođenje evidencija, administraciju podataka o igraonicama, kao i djeljenje istog servera za skladištenje podataka između više igraonica i njihovih poslovnica. |
| *Za razliku od* | dosadašnjeg načina vođenja evidencije i održavanja baze podataka |
| *Naš proizvod* | obezbjeđuje za zaposlenike: dodavanje igrica na specifični računar, brisanje igrica, zauzimanje računara, naplata računa, čuvanje računa, pretraga računara, ispis osnovnih informacija o računaru; za administratore: odobravanje pristupa sistemu zaposlenom, izmjene podataka o zaposlenom, brisanje zaposlenog, dodavanje računara, izmjene podataka o računaru, uklanjanje računara. |

## Perspektiva aplikacije

Informacioni sistem rađen je u ORACLE ovoj bazi podataka MySQL. Prikaz baze podataka je rađen prema funkcionalnim zahtjevima baze podataka, dok aplikacija urađena nad informacionim sistemom omogućava interakciju između korisnika i podataka.

Aplikacija je rađena na .NET platform u framework – u za WPF čija je namjena programiranje softvera sa modernim GUI-om(Grafičkim korisničkim interfejsom).

Izrada GUI – a je rađena poštovajući sve principe dizajniranja kako bi korisniku olakšalo učenje aplikacije, dok informacioni sistem nad kojim je podignuta aplikacija rađen kako bi se postigla što bolja efikasnost i olakšalo buduće održavanje.



# Opis korisnika

[To effectively provide products and services that meet your stakeholders’ and users' real needs, it is necessary to identify and involve all of the stakeholders as part of the Requirements Modeling process. You must also identify the users of the system and ensure that the stakeholder community adequately represents them. This section provides a profile of the stakeholders and users involved in the project, and the key problems that they perceive to be addressed by the proposed solution. It does not describe their specific requests or requirements as these are captured in a separate stakeholder requests artifact. Instead, it provides the background and justification for why the requirements are needed.]

## Opis potencijalnog tržišta

[Summarize the key market demographics that motivate your product decisions. Describe and position target market segments. Estimate the market’s size and growth by using the number of potential users or the amount of money your customers spend trying to meet needs that your product or enhancement would fulfill. Review major industry trends and technologies. Answer these strategic questions:

• What is your organization’s reputation in these markets?

• What would you like it to be? pis potencijalnog tržišta

• How does this product or service support your goals?]

## Profili korisnika

[There are a number of stakeholders with an interest in the development and not all of them are end users. Present a summary list of these non-user stakeholders. (The users are summarized in section 3.3.)]

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibilities** |
| [Name the stakeholder type.] | [Briefly describe the stakeholder.] | [Summarize the stakeholder’s key responsibilities with regard to the system being developed; that is, their interest as a stakeholder. For example, this stakeholder:   * ensures that the system will be maintainable * ensures that there will be a market demand for the product’s features * monitors the project’s progress * approves funding * and so forth] |

## User Summary

[Present a summary list of all identified users.]

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Responsibilities** | **Stakeholder** |
| [Name the user type.] | [Briefly describe what they represent with respect to the system.] | [List the user’s key responsibilities with regard to the system being developed; for example:   * captures details * produces reports * coordinates work * and so on] | [If the user is not directly represented, identify which stakeholder is responsible for representing the user’s interest.] |

## Opis okruženja

[Detail the working environment of the target user. Here are some suggestions:

* Number of people involved in completing the task? Is this changing?
* How long is a task cycle? Amount of time spent in each activity? Is this changing?
* Any unique environmental constraints: mobile, outdoors, in-flight, and so on?
* Which systems platforms are in use today? Future platforms?
* What other applications are in use? Does your application need to integrate with them?

This is where extracts from the Business Model could be included to outline the task and roles involved and so on.]

## Osnovne potrebe korisnika

[Describe each stakeholder in the system here by filling in the following table for each stakeholder. Remember that stakeholder types can be as divergent as users, departments, and technical developers. A thorough profile would cover the following topics for each type of stakeholder.]

### <Stakeholder Name>

|  |  |
| --- | --- |
| **Representative** | [Who is the stakeholder representative to the project? (Optional if documented elsewhere.) What we want here is names.] |
| **Description** | [A brief description of the stakeholder type.] |
| **Type** | [Qualify the stakeholder’s expertise, technical background, and degree of sophistication—that is, guru, business, expert, casual user, and so on.] |
| **Responsibilities** | [List the stakeholder’s key responsibilities with regard to the system being developed—that is, their interest as a stakeholder.] |
| **Success Criteria** | [How does the stakeholder define success?  How is the stakeholder rewarded?] |
| **Involvement** | [How is the stakeholder involved in the project? Relate where possible to Rational Unified Process roles—that is, Requirements Reviewer and so on.] |
| **Deliverables** | [Are there any additional deliverables required by the stakeholder? These could be project deliverables or outputs from the system under development.] |
| **Comments / Issues** | [Problems that interfere with success and any other relevant information go here.] |

## User Profiles

[Describe each unique user of the system here by filling in the following table for each user type. Remember user types can be as divergent as gurus and novices. For example, a guru might need a sophisticated, flexible tool with cross-platform support, while a novice might need a tool that is easy to use and user-friendly. A thorough profile needs to cover the following topics for each type of user.]

### <User Name>

|  |  |
| --- | --- |
| **Representative** | [Who is the user representative to the project? (Optional if documented elsewhere.) This often refers to the Stakeholder that represents the set of users, for example, Stakeholder: Stakeholder1.] |
| **Description** | [A brief description of the user type.] |
| **Type** | [Qualify the user’s expertise, technical background, and degree of sophistication—that is, guru, casual user, and so on.] |
| **Responsibilities** | [List the user’s key responsibilities with regard to the system being developed— that is, captures details, produces reports, coordinates work, and so forth.] |
| **Success Criteria** | [How does the user define success?  How is the user rewarded?] |
| **Involvement** | [How is the user involved in the project? Relate where possible to Rational Unified Process roles—that is, Requirements Reviewer, and so on.] |
| **Deliverables** | [Are there any deliverables the user produces and, if so, for whom?] |
| **Comments / Issues** | [Problems that interfere with success and any other relevant information go here. These would include trends that make the user’s job easier or harder.] |

## Key Stakeholder or User Needs

[List the key problems with existing solutions as perceived by the stakeholder or user. Clarify the following issues for each problem:

• What are the reasons for this problem?

• How is it solved now?

• What solutions does the stakeholder or user want?]

[It is important to understand the **relative** importance the stakeholder or user places on solving each problem. Ranking and cumulative voting techniques indicate problems that **must** be solved versus issues they would like addressed.

Fill in the following table—if using Rational RequisitePro to capture the Needs, this could be an extract or report from that tool.]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| Broadcast messages |  |  |  | |  |

## Alternative I konkurencija

[Identify alternatives the stakeholder perceives as available. These can include buying a competitor’s product, building a homegrown solution or simply maintaining the status quo. List any known competitive choices that exist or may become available. Include the major strengths and weaknesses of each competitor as perceived by the stakeholder or end user.]

### <aCompetitor>

### <anotherCompetitor>

# Opis proizvoda

[This section provides a high level view of the product capabilities, interfaces to other applications, and system configurations. This section usually consists of three subsections, as follows:

• Product perspective

• Product functions

• Assumptions and dependencies]

## Perspektiva proizvoda

[This subsection of the **Vision** document puts the product in perspective to other related products and the user’s environment. If the product is independent and totally self-contained, state it here. If the product is a component of a larger system, then this subsection needs to relate how these systems interact and needs to identify the relevant interfaces between the systems. One easy way to display the major components of the larger system, interconnections, and external interfaces is with a block diagram.]

## Mogućnosti

[Summarize the major benefits and features the product will provide. For example, a **Vision** document for a customer support system may use this part to address problem documentation, routing, and status reporting without mentioning the amount of detail each of these functions requires.

Organize the functions so the list is understandable to the customer or to anyone else reading the document for the first time. A simple table listing the key benefits and their supporting features might suffice. For example:]

**Table 4-1 Customer Support System**

|  |  |
| --- | --- |
| **Koristi korisnika** | **Funkcija proizvoda koje to omogućavaju** |
| New support staff can quickly get up to speed. | Knowledge base assists support personnel in quickly identifying known fixes and workarounds. |
| Customer satisfaction is improved because nothing falls through the cracks. | Problems are uniquely itemized, classified and tracked throughout the resolution process. Automatic notification occurs for any aging issues. |
| Management can identify problem areas and gauge staff workload. | Trend and distribution reports allow high level review of problem status. |
| Distributed support teams can work together to solve problems. | Replication server allows current database information to be shared across the enterprise. |
| Customers can help themselves, lowering support costs and improving response time. | Knowledge base can be made available over the Internet. Includes hypertext search capabilities and graphical query engine. |

## Pretpostavke i zavisnosti

[List each of the factors that affect the features stated in the **Vision** document. List assumptions that, if changed, will alter the **Vision** document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the **Vision** document will need to change.]

## Cijena

[For products sold to external customers and for many in-house applications, cost and pricing issues can directly impact the application’s definition and implementation. In this section, record any cost and pricing constraints that are relevant. For example, distribution costs, (# of diskettes, # of CD-ROMs, CD mastering) or other cost of goods sold constraints (manuals, packaging) may be material to the projects success, or irrelevant, depending on the nature of the application.]

## Licenciranje i instalacija

[Licensing and installation issues can also directly impact the development effort. For example, the need to support serializing, password security or network licensing will create additional requirements of the system that must be considered in the development effort.

Installation requirements may also affect coding or create the need for separate installation software.]

# Funkcionalni zahtjevi

[List and briefly describe the product features. Features are the high-level capabilities of the system that are necessary to deliver benefits to the users. Each feature is an externally desired service that typically requires a series of inputs to achieve the desired result. For example, a feature of a problem tracking system might be the ability to provide trending reports. As the use-case model takes shape, update the description to refer to the use cases.

Because the **Vision** document is reviewed by a wide variety of involved personnel, the level of detail needs to be general enough for everyone to understand. However, enough detail must be available to provide the team with the information they need to create a use-case model.

To effectively manage application complexity, we recommend for any new system, or an increment to an existing system, capabilities are abstracted to a high enough level so 25-99 features result. These features provide the fundamental basis for product definition, scope management, and project management. Each feature will be expanded in greater detail in the use-case model.

Throughout this section, each feature will be externally perceivable by users, operators or other external systems. These features need to include a description of functionality and any relevant usability issues that must be addressed. The following guidelines apply:

• Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented.

• If you are using the Rational RequisitePro toolkit, all need to be selected as requirements of type for easy reference and tracking.]

## <aFeature>

## <anotherFeature>

# Ograničenja

[Note any design constraints, external constraints or other dependencies.]

# Kvalitet

[Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.]

# Prioritet funkcionalnosti

[Define the priority of the different system features.]

# Nefunkcionalni zahtjevi

[At a high level, list applicable standards, hardware or platform requirements, performance requirements, and environmental requirements.]

## Standardi

[List all standards with which the product must comply. These can include legal and regulatory (FDA, UCC) communications standards (TCP/IP, ISDN), platform compliance standards (Windows, UNIX, and so on), and quality and safety standards (UL, ISO, CMM).]

## Sistemski zahtjevi

[Define any system requirements necessary to support the application. These can include the supported host operating systems and network platforms, configurations, memory, peripherals, and companion software.]

## Performanse

[Use this section to detail performance requirements. Performance issues can include such items as user load factors, bandwidth or communication capacity, throughput, accuracy, and reliability or response times under a variety of loading conditions.]

## Okruženje

[Detail environmental requirements as needed. For hardware- based systems, environmental issues can include temperature, shock, humidity, radiation, and so forth. For software applications, environmental factors can include usage conditions, user environment, resource availability, maintenance issues, and error handling and recovery.]

# Dokumentacija

[This section describes the documentation that must be developed to support successful application deployment.]

## Korisničko upustvo

[Describe the purpose and contents of the User Manual. Discuss desired length, level of detail, need for index, glossary of terms, tutorial versus reference manual strategy, and so on. Formatting and printing constraints must also be identified.]

## Online pomoć

[Many applications provide an online help system to assist the user. The nature of these systems is unique to application development as they combine aspects of programming (hyperlinks, and so forth) with aspects of technical writing, such as organization and presentation. Many have found the development of an online help system is a project within a project that benefits from up-front scope management and planning activity.]

## Upustvo za instalaciju i ReadMe fajl

[A document that includes installation instructions and configuration guidelines is important to a full solution offering. Also, a Read Me file is typically included as a standard component. The Read Me file can include a "What's New With This Release” section, and a discussion of compatibility issues with earlier releases. Most users also appreciate documentation defining any known bugs and workarounds in the Read Me file.]

## Izgled pakovanja proizvoda

[Today's state-of-the-art applications provide a consistent look and feel that begins with product packaging and manifests through installation menus, splash screens, help systems, GUI dialogs, and so on. This section defines the needs and types of labeling to be incorporated into the code. Examples include copyright and patent notices, corporate logos, standardized icons and other graphic elements, and so forth.]

# A Feature Attributes

[Features are given attributes that can be used to evaluate, track, prioritize, and manage the product items proposed for implementation. All requirement types and attributes need to be outlined in the Requirements Management Plan, however, you may wish to list and briefly describe the attributes for features that have been chosen. The following subsections represent a set of suggested feature attributes.]

## A.1 Status

[Set after negotiation and review by the project management team. Tracks progress during definition of the project baseline.]

|  |  |
| --- | --- |
| Proposed | [Used to describe features that are under discussion but have not yet been reviewed and accepted by the "official channel," such as a working group consisting of representatives from the project team, product management, and user or customer community.] |
| Approved | [Capabilities that are deemed useful and feasible, and have been approved for implementation by the official channel.] |
| Incorporated | [Features incorporated into the product baseline at a specific point in time.] |

## A.2 Benefit

[Set by Marketing, the product manager or the business analyst. All requirements are not created equal. Ranking requirements by their relative benefit to the end user opens a dialog with customers, analysts, and members of the development team. Used in managing scope and determining development priority.]

|  |  |
| --- | --- |
| Critical | [Essential features. Failure to implement means the system will not meet customer needs. All critical features must be implemented in the release or the schedule will slip.] |
| Important | [Features important to the effectiveness and efficiency of the system for most applications. The functionality cannot be easily provided in some other way. Lack of inclusion of an important feature may affect customer or user satisfaction, or even revenue, but release will not be delayed due to lack of any important feature.] |
| Useful | [Features that are useful in less typical applications will be used less frequently or for which reasonably efficient workarounds can be achieved. No significant revenue or customer satisfaction impact can be expected if such an item is not included in a release.] |

## A.3 Effort

[Set by the development team. Because some features require more time and resources than others, estimating the number of team or person-weeks, lines of code required or function points, for example, is the best way to gauge complexity and set expectations of what can and cannot be accomplished in a given time frame. Used in managing scope and determining development priority.]

## A.4 Risk

[Set by development team based on the probability the project will experience undesirable events, such as cost overruns, schedule delays or even cancellation. Most project managers find categorizing risks, as high, medium, and low, is sufficient, although finer gradations are possible. Risk can often be indirectly assessed by measuring the uncertainty (range) of the projects team’s schedule estimate.]

## A.5 Stability

[Set by the analyst and development team, this is based on the probability that features will change or the team’s understanding of the feature will change. Used to help establish development priorities and determine those items for which additional elicitation is the appropriate next action.]

## A.6 Target Release

[Records the intended product version in which the feature will first appear. This field can be used to allocate features from a **Vision** document into a particular baseline release. When combined with the status field, your team can propose, record, and discuss various features of the release without committing them to development. Only features whose Status is set to Incorporated and whose Target Release is defined will be implemented. When scope management occurs, the Target Release Version Number can be increased so the item will remain in the **Vision** document but will be scheduled for a later release.]

## A.7 Assigned To

[In many projects, features will be assigned to "feature teams" responsible for further elicitation, writing the software requirements, and implementation. This simple pull-down list will help everyone on the project team to understand responsibilities better.]

## A.8 Reason

[This text field is used to track the source of the requested feature. Requirements exist for specific reasons. This field records an explanation or a reference to an explanation. For example, the reference might be to a page and line number of a product requirement specification or to a minute marker on a video of an important customer review.]