Amperia

Specifikacija softverskih zahtjeva

Verzija 0.1

Istorija Revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 08.04.2023 | 0.1 | Početak izrade dokumenta, Unos funkcionalnih zahtjeva | Goran Milanović |
| 09.04.2023 | 0.2 | Dorada dokumenta | Goran Milanović |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Uvod 4

1.1 Svrha 4

1.2 Namjena 4

1.3 Definicije, akronimi i skraćenice 4

1.4 Reference 4

1.5 Pregled 4

2. Generalni opis 4

3. Specifični zahtjevi 5

3.1 Funkcionalni zahtjevi 5

3.1.1 <Functional Requirement One> 5

3.2 Nefunkcionalnizahtjevi 5

3.2.1 <Usability Requirement One> 5

3.3 Reliability 6

3.3.1 <Reliability Requirement One> 6

3.4 Performance 6

3.4.1 <Performance Requirement One> 6

3.5 Supportability 6

3.5.1 <Supportability Requirement One> 7

3.6 Design Constraints 7

3.6.1 <Design Constraint One> 7

3.7 On-line User Documentation and Help System Requirements 7

3.8 Purchased Components 7

3.9 Interfaces 7

3.9.1 User Interfaces 7

3.9.2 Hardware Interfaces 7

3.9.3 Software Interfaces 7

3.9.4 Communications Interfaces 7

3.10 Licensing Requirements 7

3.11 Legal, Copyright, and Other Notices 8

3.12 Applicable Standards 8

4. Supporting Information 8

Specifikacija softverskih zahtjeva

# Uvod

## Svrha

Svrha dokumenta Specifikacija softverskih zahtjeva je definisanje zahtjeva, kao i detaljan opis funkcionalnosti grafičkog editora za razvoj prostih električnih kola. Tokom životnog ciklusa „Amperia“ softvera za grafički razvoj prostih električnih kola ovaj dokument je podložan promjenama.

## Namjena

Dokument je namjenjen razvojnom timu „Amperia“ projekta kao osnova za razvoj „Amperia“ softvera grafičkog editora za prosta električna kola.

## Definicije, akronimi i skraćenice

Sve potrebne definicije nepoznatih pojmova, akronimi kao i skraćenice nalaze se u dokumentu Riječnik[1].

## References

* [1]Riječnik

## Pregled

U nastavku dokumenta Specifikacija softverskih zahtjeva nalaze se definisani funkcionalni i nefunkcionalni zahtjevi, ogranicenja mogućnosti softvera, kao i sigurnosna ograničenja softvera.

# Generalni opis

## Karakteristike korisnika

Od korisnika se očekuje da posjeduje osnovnu informatičku pismenost.

## Perspektiva proizvoda

“Amperia” grafički editor za prosta električna kola je alat koji pruža mogućnost korisnicima da vizuelno dizajniraju električna kola, postavljanje parametara električnih elemenata u tom kolu, kao i graficki prikaz simulacije rada električnih elemenata u preostom električnom kolu

# Specifični zahtjevi

Specifični zahtjevi predstavljaju jasno definisane zahtjeve koji detaljno opisuju funkcionalnost kao i ogranicenja u radu softvera.

## Funkcionalni zahtjevi

Funkcionalni zahtjevi opisuju šta se očekuje od sistema ili kako sistem treba da radi. Navedeni funkcionalni zahtjevi izvedeni su na osnovu testiranja sličnih grafičkih sistema.

### Rad sa korisničkim nalogom

#### Registracija korisničkog naloga

#### Prijava na korisnički nalog

#### Odjava sa korisničkog naloga

#### Promjena lozinke

### Rad sa projektima

#### Kreiranje novog projekta

#### Pokretanje postojećeg projekta

#### Čuvanje projekta

#### Čuvanje izmjena na projektu

#### Štampanje projekta

#### Brisanje projekta

#### Pretraga projekata po nazivu

#### Promjena naziva projekta

### Rad sa unapred generisanim osnovnim šablonima

#### Odabir unapred generisanog šablona

#### Pretraga šablona po nazivu

### Rad sa dijagramima

#### Kreiranje novog dijagrama

#### Brisanje dijagrama

#### Promjena imena dijagrama

#### Uvećanje prikaza dijagrama

#### Umanjenje prikaza dijagrama

### Rad sa označenim elemntima

#### Umetanje elemenata u dijagram

#### Brisanje elementa

#### Povezivanje elemenata

#### Grupisanje elemenata

#### Promjena karakteristika elementa

##### Promjena boje označenog elementa

##### Promjena električne veličine elementa

##### Promjena dimenzija elementa

#### Isjecanje elementa

#### Kopiranje elementa

#### Lijepljenje elementa

#### Povratak na predhodni korak

#### Povratak na poništeni korak

### Rad sa simulacijom

#### Pokretanje simulacije

#### Zaustavljanje simulacije

#### Pauziranje simlacije

#### Čuvanje podataka simulacije

### Podešavanja kanvasa

#### Postavljanje mreže ne kanvas

#### Promjena boje pozadine kanvasa

## Nefunkcionalni zahtjevi

[This section includes all those requirements that affect usability. For example,

* specify the required training time for a normal users and a power user to become productive at particular operations
* specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like
* specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]

### <Usability Requirement One>

[The requirement description goes here.]

## Reliability

[Requirements for reliability of the system should be specified here. Some suggestions follow:

* Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.
* Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.
* Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?
* Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system’s output.
* Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).
* Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]

### <Reliability Requirement One>

[The requirement description.]

## Performance

[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

* Response time for a transaction (average, maximum)
* Throughput, for example, transactions per second
* Capacity, for example, the number of customers or transactions the system can accommodate
* Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)
* Resource utilization, such as memory, disk, communications, and so forth.

### <Performance Requirement One>

[The requirement description goes here.]

## Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.]

### <Supportability Requirement One>

[The requirement description goes here.]

## Design Constraints

[This section indicates any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]

### <Design Constraint One>

[The requirement description goes here.]

## On-line User Documentation and Help System Requirements

[Describes the requirements, if any, for o-line user documentation, help systems, help about notices, and so forth.]

## Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and the like, so that the software can be developed and verified against the interface requirements.]

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this **SRS** but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so forth.]

## Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notices, wordmark, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

# Supporting Information

[The supporting information makes the **SRS** easier to use. It includes:

* Table of contents
* Index
* Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the **SRS** should explicitly state whether or not the appendices are to be considered part of the requirements.]