

Introduction of Software Requirement specification (SRS)

- Srs એ જે system develop થાય છે તેનું details માં માર્ગદર્શન આપે છે
- Srs માં project develop કરવા માટે જેટલી જરૂરિયાતો હોઈ છે તેનો સમાવેશ થાય છે.

Characteristics of a good srs document

Correct

- જો બધી જરૂરિયાતો કે જે final system માં જરૂરી છે તે દરેક SRS માં સમાવવામાં આવે તો તે SRS યોગ્ય કહેવાય છે
- યોગ્ય રીતે કરવામાં આવે છે.

Unambiguous

- SRS માં દર્શાવેલ બધી requirements નો એક અને માત્ર એક અર્થ થાતો હોય તો તે સ્પષ્ટ SRS ગણાય છે.

Complete

- SRS પૂર્ણ કહેવાય જો તેમાં બધી જરૂરિયાતો, કાર્યક્ષમતા, કામગીરી, design અને External interface નો સમાવેશ થાય.
- તે Valid અને Invalid બંને Input માટે મહત્વપૂર્ણ છે.

Consistent

- એક Consistent જરૂરિયાત બીજી કોઈ જરૂરિયાત સાથે Conflict થાતું નથી.
- તે Duplicate નથી.

Ranked for importance

- SRS એ મહત્વ નું છે જો SRS માં જરૂરિયાતો નું મહત્વ ઓળખાય જાય તો તે important છે.
- બધી જરૂરિયાતો જે Software Product સાથે સંકડાયેલી છે. તે બધી સરખી જરૂરી નથી.
- તેમાથી અમુક જરૂરી છે જ્યારે અમુક બિનજરૂરી હોઈ શકે.

Verifiable

- તે Software Product ને જરૂરિયાત ને ચકાસવા માટે કોઈ એક વ્યક્તિ અથવા મશીન થી ચેક કરી શકો છો.

Modifiable

- SRS માં ફેરફાર કરી શકાય છે, જો તેનું બંધારણ અને Style તેમાં કોઈ જરૂરિયાત હોય તો તેમાં સરળતાથી ફેરફારો કરી શકાય છે. અને તે સંપૂર્ણપણે અને સતત થાય છે.

Structured

- SRS જો Modules માં હોય તો તેને સમજવી અને બદલાવી સહેલી રહે છે.
- અમુક time પછી જો Customer ની જરૂરિયાતો બદલે તો તે Requirement Specification પણ બદલે છે.
- SRS માં જો કોઈ સુધારો કરવો હોય તો તે SRS સરખો સંગઠીત હોવો જરૂરી છે.

Traceable

- એક શોધી શકાય તેવી જરૂરિયાતથી તેમાં Unique ઓળખ અને Number હોય છે.

Requirement gathering and analysis

- Analyst (developer) માહિતી ભેગી કરીને જે જરૂરિયાત છે તે મેળવે છે. જે system વિકસાવવા માટે જરૂરી છે
- અલગ-અલગ માણસો અને documents પરથી માહિતી ભેગી કરવી ખૂબ જ અઘરી છે અને તે problem ને સરખી રીતે પહેલા સમજવો જરૂરી છે.
- જો working model હાજર માં હોય તો જરૂરિયાતો ભેગી કરવામાં મદદ થાય છે.

Studying the existing documentation

- Analyst જે system exist હોય છે તેને સંબંધિત study કરીને જરૂરિયાતો ભેગી કરે છે.
- આ document માં stakeholders અને મૂળભૂત હેતુ નો સમાવેશ થાય છે.

Interview

- બધા અલગ-અલગ categories ના users ના interview લઈને જરૂરિયાતો ને ભેગી કરવામાં આવે છે.
- For example, library management system ના software માં જે developer હોય તે library ના member's અને accountant's ના interview લઈને જરૂરિયાત ભેગી કરે છે.

Task analysis

- Software ની services ના સમૂહને task કહેવાય છે, દરેક task માં analyst અલગ-અલગ step થી user ના માર્ગદર્શન મુજબ માહિતી મેળવે છે.
- For example, book issue service ના step નીચે મુજબ છે.
 - ✓ Authenticate user
 - ✓ Book ની maximum limit કે જે user લઈ શકે છે તે પહોંચી ગઈ છે કે નહીં.
 - ✓ ચેક કરો કે book reserved છે કે નહીં
 - ✓ Users ના record માં તેની book issue ની details post કરો અને છેલ્લે તેની issue slips print કરાવો.

Scenario analysis

- કાર્ય operation ના ઘણા scenario માં હોઈ શકે છે.
- એક કાર્ય ના વિવિધ પ્રકાર અને system નું વર્તન અલગ હોય છે.
- For example, અલગ-અલગ scenario book issue માટે હોય છે:
 - ✓ Book issue success થઈ જાય તો તેની book issue slips print થઈ જાવી જોઈએ.
 - ✓ જો book reserved હોય તો તે member ને issue થાતી નથી.
 - ✓ જો user ની book લેવાની maximum limit પૂરી થઈ ગઈ હોય તો તે પછી તેને issue કરી શકાતી નથી.

Form Analysis

- વિવિધ forms એ system માટેની જે ઈનપૂટ માહિતી અને system પાસેથી કઈ આઉટપુટ માહિતી મળે છે તે નક્કી કરે છે.
- Example: Student course Registration form.

Requirements analysis

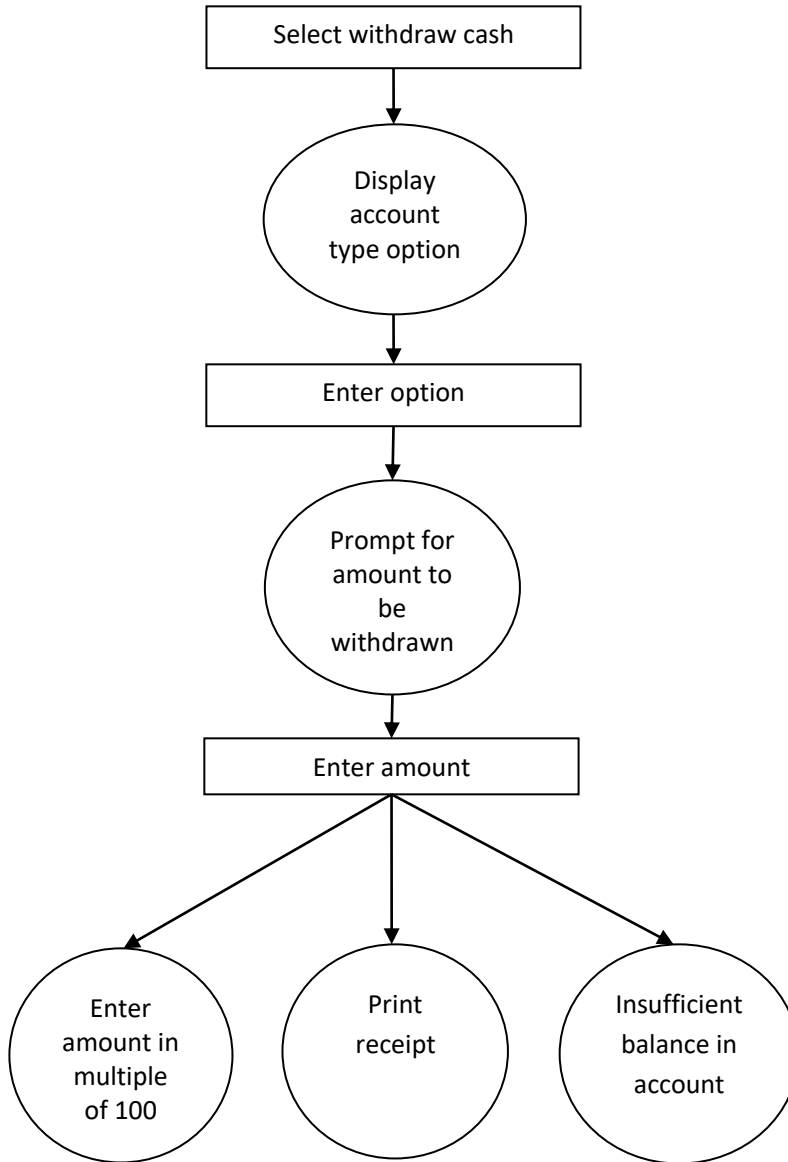
- Requirement ભેગી કર્યા પછી analyst તે ભેગી કરેલી જરૂરિયાતોમાંથી customer ની જરૂરિયાતો ને સમજે છે.
- Requirement analysis નો મુખ્ય હેતુ જે product ને બનાવવાની હોય છે તેને સરખી રીતે સમજવી, તેમાંથી ભૂલો સુધારવાનો હોય છે.
- નીચેના question નો જે analyst ને સરખા સમજવા જોઈએ:
 - ✓ Problem શું છે.?
 - ✓ તે problem ને હલ કરવો શા માટે જરૂરી છે. ?
 - ✓ તે problem માટેના શક્ય એવા solution શું છે?
 - ✓ ક્યો data input માં માપવામાં આવે છે અને system પાસેથી output માં શું મળે છે. ?
 - ✓ Problem Solve કરી ત્યારે કેવા complex હશે?
- જો analysis તેમાં અપૂર્ણતા અથવા તો ભૂલો ને શોધે છે. ત્યારે તેમણે customer સાથે ચર્ચાઓ કરીને તેને સુધારે છે.

Functional requirements and non-functional requirements

- Requirement specification એ એવી Activity છે, જે Analysis phase માં ભેગી કરેલી માહિતીને Document માં translate કરે છે.
- તેના two types છે:
 1. Functional requirements
 2. Nonfunctional requirement

Functional Requirements

- એવી જરૂરિયાત જે System ને શું કરવું જોઈએ એ સ્પષ્ટ કરે છે.
- Functional Requirement System ની સ્થિતિ દર્શાવે છે.
- Example: Email મોકલવો જ્યારે કોઈ નવો user sign up કરે અથવા નવું Account ખોલવા માટે Request કરે.
- Functional Requirement System પાસેથી કેવી અપેક્ષા રાખે છે તેની ચર્ચા કરે છે. આ System કોઈ ખાસ Input કેવી રીતે વર્તન જોઈએ અને Particular પરિસ્થિતી માં કેવું વર્તન કરવું જોઈએ , તે Input અને Output વચ્ચેનો સંબંધ વર્ણવે છે
- System high level ના function કરવા માટે ઉપયોગ કરવામાં આવે છે.
- System ના દરેક કાર્ય Input ના set ને Output પ્રમાણે Transform કરે છે.



- પહેલા System ના High Level Function ને ઓળખો
- High Level ના Function ને નાની Sub જરૂરિયાત માં વિભાજીત કરવામાં આવશે.
- High Level Function ના ઉપયોગથી User કેટલાક ઉપયોગી કામ કરી શકે છે.
- For Example, જ્યારે ATM માંથી પૈસા ઉપાડી ત્યારે રસીદ Print કામ ઉપયોગી કહેવાય છે? Receipt Printing કામ ને High Level ની જરૂરિયાત ગણવી જોઈએ નહીં, પરંતુ User એ ખાસ આ પ્રવૃત્તિ માટે Request કરી નથી કારણકે Receipt Automatic Print થઈ જાય છે. તેને પૈસા ઉપાડવાનો એક Part ગણી શકાય છે.
- Library Management System ની High Level Functional Requirement Book ને Search કરવી એ છે. આ કાર્ય માં Book નું નામ અથવા Book ના List માં Algorithm Match કરીને છેલ્લે મેળ ખાતી Book Output માં આવે છે.
- High Level Function માં સામાન્ય રીતે System અને User વચ્ચે Interaction કરવામાં આવે છે.

- For Example, આકૃતિમાં દર્શાવ્યા પ્રમાણે Input એ Rectangle અને System નો Response Circle દ્વારા દર્શાવવામાં આવે છે.

Non-functional requirements

- એવી જરૂરિયાતો છે જે System કેવી રીતે ચોક્કસ કાર્ય કરે છે તે સ્પષ્ટ કરે છે.
- Non-Functional જરૂરિયાતો માં કેવી રીતે System વર્તન કરશે અને તેમાં શું Limits હશે તેની કામગીરી ઉપર હોય છે.
- Non-Functional Requirement System ની લાક્ષણિકતા કે જે Function માં વ્યક્ત કરી શકતી નથી જેવી કે System ની Maintainability, System ની ઉપયોગિતા, તેમાં Maximum કેટલા Current User હશે તે દર્શાવે છે.
- Non-Functional Requirement ની અંદર ,
 - ✓ reliability issues
 - ✓ accuracy of results
 - ✓ Constraints on the system implementation, etc.
- Non-Functional નું example આપણે લઈ શકીએ કે સોફ્ટવેર નું user interface factory workerwork shop માં વર્ક કરતા worker પણ use કરી સકે તેવું હોવું જોઈએ જેમની પાસે degree નથી.

Example 1 : Online shopping system

Functional requirements એવી જરૂરિયાત જે System ને શું કરવું જોઈએ એ સ્પષ્ટ કરે છે.

- Maintenance:** product ની detail અને તેના attributes જેમ કે size, weight, cost, વગેરે maintain કરે છે.
- Product Management:** તેમાં information આઈટમ નંબર, size, categories વગેરેનો સમાવેશ થાય છે.
- Stock report:** quantity product ની અને product નું status ને indicate કરે છે.
- Registration:** જો customer ને product purchase કરવી હોઈ તો તેને registraion કરવું ફરજિયાત છે. register નો કરેલ customer shopping cart માં product purchase કરી શકે નહિ.
- Login:** customer valid username અને password વડે સિસ્ટમ માં લોગીન કરી શકે.
- View item:** જે available માં હશે તે product ને customer category wise જોઈ શકે.
- Order item:** customer product ને shopping cart ની અંદર add કરી શકે અને તે order ને shppoing cart માંથી cancel પણ કરી શકે.
- Select payment mode:** Customer payment કેશ, credit કાર્ડ, ડેબીટ કાર્ડ, online વગેરે થી કરી શકે.
- Logout:** Order થઈ ગયા પછી customer લોગઆઉટ થઈ શકે છે.

Non functional requirements

- Usability:** website non-technical વ્યક્તિ માટે પણ વાપરવામાં સરળ હોવી જોઈએ.
- Security:** જો સાઈટ નાણાકીય વ્યવહારો, user ના નાણાકીય અને Sensitive ડેટા સાથે કામ કરતી હોય તો Security સૌથી મહત્વપૂર્ણ છે. Data privacy user ને સાઈટ માટે નો trust બનાવી રાખશે.
- Performance:** વેબસાઈટ પર ટ્રાફિક વધારવા માટે, તમારે non-financial documentation કામગીરી પર

વિશેષ ધ્યાન આપવું પડશે.

- **Safety Requirements:** વાયરસ અથવા ઓપરેટિંગ સિસ્ટમની failure ને કારણે ડેટાબેઝ કોઈપણ ચોક્કસ સમયે ક્રેશ થઈ શકે છે. તેથી, ડેટાબેઝ બેકઅપ લેવો જરૂરી છે.

Example 2 : LMS Student Portal

Functional requirements

- **Login:** customer valid username અને password વડે સિસ્ટમ માં લોગીન કરી શકસે.
- **View / Edit profile:** user તેની પ્રોફાઈલ view અને edit બંને કરી શકે છે.
- **Select Semester:** સેમેસ્ટર filed માંથી User Semester ને select કરી શકે છે.
- **View Subject list:** User particular semester wise વિષય લીસ્ટ જોઈ શકે છે અને syllabus જોઈ અને download કરી શકે છે.
- **Download e-notes Presentation:** particular સેમેસ્ટર અને subject wise યુઝર e-notes અને presentation download કરી શકે છે.
- **View class time-table:** Student અહિયાં થી timetable જોઈ અને download કરી શકે છે.
- **View Exam time table and schedule:** Exam time એ user અહિયાં થી તેનું exam time-table જોઈ અને download કરી શકે છે.
- **Give test:** particular વિષય wise લેવામાં આવતી test યુઝર Give test Filed પર આપે છે.
- **View result:** student તેનું result ને view જોઈ શકે છે.
- **View lab/Lecture Attendance:** user તેની attendance lab/lecture ની percentage wise જોઈ શકે છે.
- **Give Feedback:** overall college campus, practical lab, subject faculty પ્રમાણે user feedback આપી શકે છે.
- **Forget password:** જો user ને તેનો password ભૂલાય ગયો હોઈ તો તે password forgot કરી ને પાછો મેળવી શકે છે.
- **Logout:** profile માં work પૂરું થઈ ગયા પછી user તેની profile logout કરી સકે છે.

Non-functional requirements

- **Usability:** website non-technical વ્યક્તિ માટે પણ વાપરવામાં સરળ હોવી જોઈએ.
- **Security:** જો સાઈટ નાણાકીય વ્યવહારો, user ના નાણાકીય અને Sensitive ડેટા સાથે કામ કરતી હોય તો Security સૌથી મહત્વપૂર્ણ છે. Data privacy user ને સાઈટ માટે નો trust બનાવી રાખશે.
- **Performance:** વેબસાઈટ પર ટ્રાફિક વધારવા માટે, તમારે non-financial documentation કામગીરી પર વિશેષ ધ્યાન આપવું પડશે.
- **Safety Requirements:** વાયરસ અથવા ઓપરેટિંગ સિસ્ટમની failure ને કારણે ડેટાબેઝ કોઈપણ ચોક્કસ સમયે ક્રેશ થઈ શકે છે. તેથી, ડેટાબેઝ બેકઅપ લેવો જરૂરી છે.