

Subject Name (Gujarati)

4353202 -- Summer 2025

Semester 1 Study Material

Detailed Solutions and Explanations

પ્રશ્ન 1(અ) [3 ગુણ]

બધા જ પ્રકારના સોફ્ટવેર એપ્લિકેશન ડોમેઇન ની યાદી બનાવો અને Embedded Software સમજાવો

જવાબ

સોફ્ટવેર એપ્લિકેશન ડોમેઇન:

ડોમેઇન	વર્ણન
સિસ્ટમ સોફ્ટવેર	ઓપરેટિંગ સિસ્ટમ, ડિવાઇસ ફ્રાઇવર
એપ્લિકેશન સોફ્ટવેર	વર્ડ પ્રોસેસર, ગેમ્સ, બિજનેસ એપ્સ
ઓન્જિનિયરિંગ/સાયન્ટિફિક સોફ્ટવેર	CAD, સિમ્યુલેશન ટૂલ
એમ્બેડેડ સોફ્ટવેર	રિયલ-ટાઈમ કંટ્રોલ સિસ્ટમ
વેબ એપ્લિકેશન	બ્રાઉઝર-આધારિત એપ્લિકેશન
AI સોફ્ટવેર	મશીન લર્નિંગ, એક્સપર્ટ સિસ્ટમ

એમ્બેડેડ સોફ્ટવેર એ વિશેષ સોફ્ટવેર છે જે ચોક્કસ હાર્ડવેર સાથે એમ્બેડેડ સિસ્ટમ પર ચાલે છે. આ વોશિંગ મશીન, કાર અને મેડિકલ ઉપકરણોમાં વપરાય છે.

- રિયલ-ટાઈમ ઓપરેશન: નિર્ધારિત સમયમર્યાદામાં જવાબ આપવો જોઈએ
- રિસોર્સ મર્યાદાઓ: મર્યાદાએ મેમરી અને પ્રોસેસિંગ પાવર
- હાર્ડવેર પર નિર્ભરતા: ચોક્કસ હાર્ડવેર સાથે ગાઢ એકીકરણ

મેમરી ટ્રીક

"SAEEWA" - System, Application, Engineering, Embedded, Web, AI

પ્રશ્ન 1(બ) [4 ગુણ]

જેનેરિક ફેમવર્ક એક્ટિવિટીસ અને અમ્બ્રેલા એક્ટિવિટીસ સમજાવો

જવાબ

જેનેરિક ફેમવર્ક એક્ટિવિટીસ:

એક્ટિવિટી	હેતુ
કોમ્પ્યુનિકેશન	હિતધારકોથી જરૂરિયાતો એકત્રિત કરવી
લ્યાન્ચિંગ	કાર્ય યોજના અને શેડ્યુલ બનાવવું
મોડેલિંગ	વિશ્લેષણ અને ડિઝાઇન મોડેલ બનાવવા
કન્સ્ટ્રક્શન	કોડ જનરેશન અને ટેસ્ટિંગ
ડિફ્લોયરેન્ટ	સોફ્ટવેર ડિલિવરી અને સપોર્ટ

અમ્બેલા એક્ટિવિટીસ:

એક્ટિવિટી	હેતુ
પ્રોજેક્ટ મેનેજમેન્ટ	પ્રગતિ ટ્રેક કરવી અને નિયંત્રણ
રિસ્ક મેનેજમેન્ટ	જોખમો ઓળખવા અને ઘટાડવા
ક્વોલિટી એસ્થોર્નસ	સોફ્ટવેર ગુણવત્તા સુનિશ્ચિત કરવી
કન્ફિગરેશન મેનેજમેન્ટ	ફેરફારોને નિયંત્રિત કરવા
વર્ક પ્રોડક્ટ પ્રિપરેશન	દસ્તાવેજુકરણ બનાવવું

- ફેમવર્ક એક્ટિવિટીસ: દરેક પ્રોજેક્ટમાં મુખ્ય ક્રમિક પ્રવૃત્તિઓ
- અમ્બેલા એક્ટિવિટીસ: પ્રોજેક્ટ જીવનકાળ દરમિયાન સતત પ્રવૃત્તિઓ

મેમરી ટ્રીક

“CPMCD” ફેમવર્ક માટે, “PRQCW” અમ્બેલા માટે

પ્રશ્ન 1(ક) [7 ગુણ]

સોફ્ટવેર ડેવલપમેન્ટ લાઇફ સાઇકલની આકૃતિ દોરી તેના તબક્કાઓ સમજાવો

જવાબ

SDLC આકૃતિ:

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    A[Requirements Analysis] --> B[System Design]
    B --> C[Implementation]
    C --> D[Testing]
    D --> E[Deployment]
    E --> F[Maintenance]
    F --> A
```

{Highlighting}

{Shaded}

SDLC તબક્કાઓ:

તબક્કો	પ્રવૃત્તિઓ	પરિણામો
જરૂરિયાત વિશ્લેષણ	વપરાશકર્તા જરૂરિયાતો એકનિત કરવી, SRS બનાવવું	SRS દસ્તાવેજ
સિસ્ટમ ડિઝાઇન	આર્કિટેક્ચર ડિઝાઇન, UI ડિઝાઇન	ડિઝાઇન દસ્તાવેજ
અમલીકરણ	કોડ ડેવલપમેન્ટ, યુનિટ ટેસ્ટિંગ	સોર્સ કોડ
ટેસ્ટિંગ	એકીકરણ, સિસ્ટમ ટેસ્ટિંગ	ટેસ્ટ રિપોર્ટ
ડિપ્લોયમેન્ટ	ઇન્સ્ટોલેશન, વપરાશકર્તા તાલીમ	ડિપ્લોય થયેલ સિસ્ટમ
જાળવણી	બગ ફિક્સ, સુધારાઓ	અપડેટ થયેલ સિસ્ટમ

- વ્યવસ્થિત અભિગમ: દરેક તબક્કાના ચોક્કસ ઇનપુટ અને આઉટપુટ
- ગુણવત્તા ગેટ: તબક્કાઓ વચ્ચે સમીક્ષા ગુણવત્તા સુનિશ્ચિત કરે છે
- પુનરાવર્તિત પ્રકૃતિ: પ્રતિપુષ્ટિ આગામી ચક્કો સુધારે છે

મેમરી ટ્રીક

“વાસ્તવિક સિસ્ટમ અમલીકરણ ટેસ્ટ દરમિયાન જાળવણી”

પ્રશ્ન 1(ક) OR [7 ગુણ]

સોફ્ટવેર ડેવલપમેન્ટ મોડેલ્સની ચાદી બનાવી કોઈ પણ બે મોડલ જરૂરી આફુતિ સાથે સમજાવો

જવાબ

સોફ્ટવેર ડેવલપમેન્ટ મોડેલ્સ:

મોડેલ	લાક્ષણિકતાઓ
વોટરફોલ મોડેલ	કભિય અભિગમ
પુનરાવર્તિત મોડેલ	ડેવલપમેન્ટના પુનરાવર્તિત ચક્કો
સ્પાઇરલ મોડેલ	જોખમ-સંચાલિત, પુનરાવર્તિત
એજાઇલ મોડેલ	લવરીક, ગ્રાહક સહયોગ
RAD મોડેલ	જડપી પ્રોટોટાઇપિંગ
V-મોડેલ	વેરિફિકેશન અને વેલિડેશન પર ધ્યાન

1. વોટરફોલ મોડેલ:

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    A[Requirements] --> B[Design]
    B --> C[Implementation]
    C --> D[Testing]
    D --> E[Deployment]
    E --> F[Maintenance]
{Highlighting}
{Shaded}
```

2. સ્પાઇરલ મોડેલ:

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    A[Planning] --> B[Risk Analysis]
    B --> C[Engineering]
    C --> D[Evaluation]
    D --> A
```

- વોટરફોલ: સરળ, સારી રીતે સમજાયેલ જરૂરિયાતો માટે યોગ્ય
- સ્પાઇરલ: ઉચ્ચ જોખમવાળા પ્રોજેક્ટને પુનરાવર્તિત જોખમ મૂલ્યાંકન સાથે હેન્ડલ કરે છે

મેમરી ટ્રીક

“WIRRRAV” - Waterfall, Iterative, RAD, Risk-driven, Agile, V-model

પ્રશ્ન 2(અ) [3 ગુણ]

SCRUM એજાઇલ પ્રોસેસ મોડલ અને SPIRAL પ્રોસેસ મોડલ વચ્ચેના તફાવત લખો

જવાબ

પાસું	SCRUM	SPIRAL
અભિગમ	એજાઇલ, પુનરાવર્તિત	જોખમ-સંચાલિત, પુનરાવર્તિત

અવધિ	નિશ્ચિત સ્પ્રેન્ટ (2-4 અઠવાડિયા)	ચલ સ્પાઇરલ ચક્કો
દ્યાન	ગ્રાહક સહયોગ	જોખમ વ્યવસ્થાપન
આપોજન	સ્પ્રેન્ટ પ્લાનિંગ	વ્યાપક આપોજન
દસ્તાવેજુકરણ	ન્યૂનતમ દસ્તાવેજુકરણ	વિગતવાર દસ્તાવેજુકરણ
ટીમ સાઇઝ	નાની ટીમ (5-9 સભ્યો)	કોઈપણ ટીમ સાઇઝ

- SCRUM: જડપી ડિલિવરી અને ગ્રાહક પ્રતિપુષ્ટિ પર ભાર
- SPIRAL: જોખમ ઓળખ અને શમન પર દ્યાન

મેમરી ટ્રીક

"SCRUM=સ્પીડ, SPIRAL=સેફટી"

પ્રશ્ન 2(બ) [4 ગુણ]

જરૂરિયાત એક્ટ્રીકરણ તકનીકોની યાદી આપો અને કોઈ પણ એક સમજાવો

જવાબ

જરૂરિયાત એક્ટ્રીકરણ તકનીકો:

તકનીક	વર્ણન
ઇન્ટરવ્યુ	હિતધારકો સાથે સીધી વાતચીત
પ્રશ્નાવલી	માળખાગત લેખિત પ્રશ્નો
અવલોકન	વપરાશકર્તાઓને કાર્ય કરતા જોવા
દસ્તાવેજ વિશ્વેષણ	હાલના દસ્તાવેજોની સમીક્ષા
પ્રોટોટાઇપિંગ	કાર્યશીલ મોડેલ બનાવવા
બ્રેઇનસ્ટોમિંગ	ગૂપ આઇડિયા જનરેશન

ઇન્ટરવ્યુ તકનીક સમજાવેલ:

- માળખાગત ઇન્ટરવ્યુ: પૂર્વનિર્ધારિત પ્રશ્નો, ઔપચારિક અભિગમ
- અમાળખાગત ઇન્ટરવ્યુ: ખૂલ્લી ચર્ચા, લવચીક
- અર્ધ-માળખાગત: બંનેનું મિશ્રણ

ફાયદાઓ: સીધી હિતધારક ઇનપુટ, સ્પષ્ટીકરણ શક્ય, વિગતવાર માહિતી પડકારો: સમય વપરાશ, ઇન્ટરવ્યુઅર પૂર્વગઢ, અધૂરી માહિતી

મેમરી ટ્રીક

"IQDPBB" - Interview, Questionnaire, Document, Prototype, Brainstorm, Observe

પ્રશ્ન 2(ક) [7 ગુણ]

ચુઝ કેસ ડાયગ્રામ વ્યાખ્યાપિત કરો. તેને ઉદાહરણ સાથે સમજાવો

જવાબ

ચુઝ કેસ ડાયગ્રામ વ્યાખ્યા: ચુઝ કેસ ડાયગ્રામ એક્ટર્સ અને તેમની ચુઝ કેસ સાથેની કિયાપ્રતિક્રિયા દર્શાવીને સિસ્ટમની કાર્યતમક જરૂરિયાતો બતાવે છે.

ઘટકો:

ઘટક	પ્રતીક	હેતુ
એક્ટર	લાકડી આફુતિ	બાધ્ય એન્ટિટી
ચુઝ કેસ	અંડાકાર	સિસ્ટમ ફૂંકશન
એસોસિએશન	લાઇન	એક્ટર-ચુઝ કેસ સંબંધ
સિસ્ટમ બાઉન્ડરી	લંબચોરસ	સિસ્ટમ સ્કોપ

ઉદાહરણ: લાઇબ્રેરી મેનેજમેન્ટ સિસ્ટમ

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph TD
    A[ ] --- B()
    A --- C()
    A --- D()
    E[ ] --- B()
    E --- C()
    E --- F()
{Highlighting}
{Shaded}
```

સંબંધો:

- Include:** યુઝ કેસ દ્વારા શેર કરાયેલ સામાન્ય કાર્યક્ષમતા
- Extend:** બેઝ યુઝ કેસમાં વૈકલ્પિક કાર્યક્ષમતા ઉમેરવી
- સામાન્યીકરણ:** એકટર્સ અથવા યુઝ કેસ વરચે વારસો

ફુયદાઓ: સ્પષ્ટ કાર્યોત્કર્ષ જાંખી, ક્રમચિકિત્શન ટૂલ, ટેસ્ટિંગ માટે આધાર

મેમરી ટ્રીક

"એકટર્સ યુઝ કેસ સિસ્ટમની અંદર"

પ્રશ્ન 2(અ) OR [3 ગુણ]

વોટર ફોલ મોડલ અને ઇટરેટિવ વોટર ફોલ મોડલ ની સરખામણી કરો

જવાબ

પાસું	વોટરફોલ મોડલ	ઇટરેટિવ વોટરફોલ
તબક્કાઓ	કમિક, એક વખત	પુનરાવર્તનમાં પુનરાવૃત્તિ
પ્રતિપુષ્ટિ	પ્રોજેક્ટના અંતે	દરેક પુનરાવર્તન પછી
જોખમ	મોડેથી જોખમ ઓળખ	વહેલી જોખમ ઓળખ
લવચીકરણ	કઠોર, કોઈ ફેરફાર નહીં	ફેરફારોને સમાવે છે
ટેસ્ટિંગ	ડેવલપમેન્ટ પછી	સતત ટેસ્ટિંગ
ડિલિવરી	એક અંતિમ ડિલિવરી	બહુવિધ વૃદ્ધિશીલ ડિલિવરી

- વોટરફોલ:** સ્થિર, સારી રીતે વ્યાખ્યાપિત જરૂરિયાતો માટે યોગ્ય
- ઇટરેટિવ વોટરફોલ:** પ્રતિપુષ્ટિ સાથે વિકસિત જરૂરિયાતો માટે બહેતર

મેમરી ટ્રીક

"PFRTFD" - Phases, Feedback, Risk, Testing, Flexibility, Delivery

પ્રશ્ન 2(બ) OR [4 ગુણ]

ફૂંકશનલ અને નોન-ફૂંકશનલ જરૂરિયાતની વ્યાખ્યા લખી બંનેના ઉદાહરણ આપો

જવાબ

ફૂંકશનલ જરૂરિયાતો: સિસ્ટમે શું કરવું જોઈએ - ચોક્કસ વર્તણૂકો અને કાર્યોને વ્યાખ્યાપિત કરતી જરૂરિયાતો.
નોન-ફૂંકશનલ જરૂરિયાતો: સિસ્ટમ કેવી રીતે કાર્ય કરે છે - ગુણવત્તા લક્ષણો અને મર્યાદાઓને વ્યાખ્યાપિત કરતી જરૂરિયાતો.

પ્રકાર	ફુંક્શનલ	નોન-ફુંક્શનલ
વ્યાખ્યા	સિસ્ટમ વર્તણુક	સિસ્ટમ ગુણવત્તા
ઉદાહરણો	લોગિન, ગણતરી, સંગ્રહ	પ્રદર્શન, સુરક્ષા
ટેસ્ટિંગ	બ્લેક-બોક્સ ટેસ્ટિંગ	લોડ, સ્ટ્રેસ ટેસ્ટિંગ
દસ્તાવેજુકરણ	યુઝ કેસ, દશ્યો	ગુણવત્તા મેટ્રિક્સ

ફુંક્શનલ ઉદાહરણો:

- વપરાશકર્તા પ્રમાણીકરણ અને લોગિન
- કુલ બિલ રકમની ગણતરી કરવી
- માસિક રિપોર્ટ જનરેટ કરવી

નોન-ફુંક્શનલ ઉદાહરણો:

- સિસ્ટમ રિસ્પોન્સ ટાઈમ < 2 સેકન્ડ (પ્રદર્શન)
- 99.9% સિસ્ટમ ઉપલબ્ધતા (વિશ્વસનીયતા)
- 1000 સમવર્તી વપરાશકર્તાઓને સપોર્ટ (સ્કેલેબિલિટી)

મેમરી ટ્રીક

"ફુંક્શનલ=શું, નોન-ફુંક્શનલ=કેવી રીતે"

પ્રશ્ન 2(ક) OR [7 ગુણ]

કોહેશનની વ્યાખ્યા આપો. કોહેશનનું વર્ગીકરણ સમજાવો

જવાબ

કોહેશન વ્યાખ્યા: કોહેશન માપે છે કે મોડ્યુલની અંદરના તત્વો કેટલા નજીકથી સંબંધિત છે. ઉચ્ચ કોહેશન સારી રીતે ડિઝાઇન કરાયેલ મોડ્યુલ દર્શાવે છે.

કોહેશનનું વર્ગીકરણ (સૌથી મજબૂતથી સૌથી નબળું):

પ્રકાર	વર્ણન	ઉદાહરણ
ફુંક્શનલ	એક, સારી રીતે વ્યાખ્યાપિત કાર્ય	વર્ગમૂળ ગણતરી
સિકવણ્ણિયલ	એકનું આઉટપુટ = બીજાનું ઇનપુટ	વાંચવુંકરણ
કોમ્યુનિકેશનલ	સમાન ડેટા પર કામ કરવું	ગ્રાહક રેકોર્ડ અપડેટ
પ્રોસીજરલ	અમલીકરણનો કમ અનુસરવો	પેરોલ પ્રોસેસિંગ સ્ટેપ્સ
ટેમ્પોરલ	સમાન સમયે અમલ	સિસ્ટમ પ્રારંભીકરણ
લોજિકલ	સમાન લોજિકલ ફુંક્શન	બધા ઇનપુટ/આઉટપુટ ઓપરેશન
કોઇન્ડિનટલ	કોઈ અર્થપૂર્ણ સંબંધ નહીં	રેન્ડમ યુટિલિટીઝ

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    A[-] --- B[-]
    B --- C[-]
    C --- D[-]
    D --- E[-]
    E --- F[-]
    F --- G[-]
{Highlighting}
{Shaded}
```

લક્ષ્ય: જાળવણીયોગ્ય, વિશ્વસનીય મોડ્યુલ માટે ફુંક્શનલ કોહેશન હાંસલ કરવું

મેમરી ટ્રીક

"ફુંક્શની સ્માર્ટ બિલાડી ટેનિસ લોજિકલી રૂમે છે"

પ્રશ્ન 3(અ) [3 ગુણ]

સારા સોફ્ટવેર ડિઝાઇનની લાક્ષણિકતાઓની યાદી બનાવો

જવાબ

સારા સોફ્ટવેર ડિઝાઇનની લાક્ષણિકતાઓ:

લાક્ષણિકતા	વર્ણન
મોડ્યુલારિટી	સ્વતંત્ર મોડ્યુલમાં વિભાજિત
એવ્સ્ટ્રેક્શન	અમલીકરણ વિગતો છુપાવવી
ઓન્કેપ્સ્યુલેશન	ડેટા અને મેથડ્સ એક્સાથે બંદલ કરવા
હાયરાર્કો	સ્તરો/લેવલમાં સંગઠિત
સરળતા	સમજવામાં અને જાળવવામાં સરળ
લવ્યેક્ટા	ભવિષ્યના ફેરફારોને સમાવવા

- ઉચ્ચ કોહેશન: સંબંધિત તત્ત્વો એક્સાથે જૂથબદ્ધ
- નીચું કપલિંગ: મોડ્યુલ વર્ચે ન્યૂનતમ નિભરતાઓ
- પુનઃઉપયોગિતા: ઘટકોને અન્ય સિસ્ટમમાં ફ્રીથી વાપરી શકાય

મેમરી ટ્રીક

“MAEHSF” - Modularity, Abstraction, Encapsulation, Hierarchy, Simplicity, Flexibility

પ્રશ્ન 3(બ) [4 ગુણ]

ઈટરમીડીયેટ COCOMO મોડલ દ્વારા પ્રોજેક્ટ એસ્ટીમેશન પદ્ધતિ સમજાવો

જવાબ

ઇન્ટરમીડીયેટ COCOMO મોડલ: ઉત્પાદકતાને અસર કરતા કોસ્ટ ડ્રાઇવરોને ધ્યાનમાં લઈને બેઝિક COCOMO ને વિસ્તૃત કરે છે.

સૂત્ર: $Effort = a \times (KLOC)^b \times EAF$

કોસ્ટ ડ્રાઇવર્સ:

કેટેગરી	ડ્રાઇવર્સ	પ્રભાવ
પ્રોડક્ટ	વિશ્વસનીયતા, જટિલતા	પ્રયત્ન ગુણાક
હાઇવર	એક્ઝિઅક્યુશન ટાઇમ, સ્ટોરેજ	પ્રદર્શન મર્યાદાઓ
કર્મચારીવર્ગ	વિશ્લેષક ક્ષમતા, અનુભવ	ટીમ કુશળતા
પ્રોજેક્ટ	આધુનિક પ્રથાઓ, શેડ્યુલ	દેવલપમેન્ટ વાતાવરણ

પ્રયત્ન સમાયોજન ફેક્ટર (EAF): $EAF = બધા કોસ્ટ ડ્રાઇવર ગુણકોનું ગુણાકાર પગલાં:$

- KLOC (કોડની હજારો લાઇન) નો અંદાજ કાઢવો
- પ્રોજેક્ટ પ્રકાર આધારે યોગ્ય a, b મૂલ્યો પરસ્પંદ કરવા
- કોસ્ટ ડ્રાઇવર્સનું મૂલ્યાંકન (સ્કેલ 0.70 થી 1.65)
- EAF ની ગણતરી કરવી
- પર્સન-મંથમાં પ્રયત્ન મેળવવા માટે સૂત્ર લાગુ કરવું

મેમરી ટ્રીક

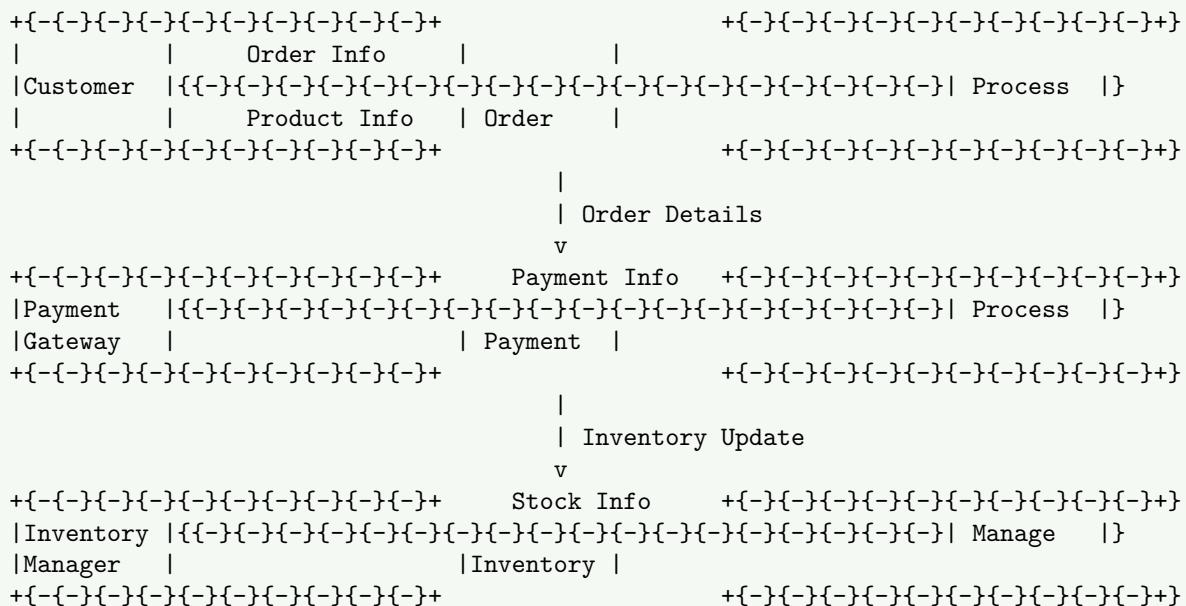
“PHPP” - Product, Hardware, Personnel, Project drivers

પ્રશ્ન 3(ક) [7 ગુણ]

ઓનલાઇન શોપિંગ સિસ્ટમ માટે લેવલ-1 નો ડેટા ફલો ડાયગ્રામ દોરો અને સમજાવો

જવાબ

ઓનલાઇન શોપિંગ સિસ્ટમ માટે લેવલ-1 DFD:



પ્રોસેસ:

પ્રોસેસ	ઇનપુટ	આઉટપુટ	વર્ણન
ઓર્ડર પ્રોસેસ	ગ્રાહક ઓર્ડર	ઓર્ડર પુષ્ટિકરણ	ઓર્ડર પ્લેસમેન્ટ હેન્ડલ કરવું
પેમેન્ટ પ્રોસેસ	પેમેન્ટ વિગતો	પેમેન્ટ સ્ટેટસ	ટ્રાન્ઝેક્શન પ્રોસેસ કરવા
ઇન્વેન્ટરી મેનેજ	સ્ટોક કવરી	સ્ટોક સ્ટેટસ	પ્રોડક્ટ ઉપલબ્ધતા ટ્રેક કરવી

ડેટા સ્ટોર:

- પ્રોડક્ટ ડેટાબેઝ: પ્રોડક્ટ માહિતી સંગ્રહિત કરવી
- ઓર્ડર ડેટાબેઝ: ઓર્ડર વિગતો સંગ્રહિત કરવી
- ગ્રાહક ડેટાબેઝ: ગ્રાહક પ્રોફાઇલ સંગ્રહિત કરવી

બાધ્ય એન્ટીટીઝ:

- ગ્રાહક: ઓર્ડર મૂકે છે, પેમેન્ટ કરે છે
- પેમેન્ટ ગેટવે: પેમેન્ટ પ્રોસેસ કરે છે
- ઇન્વેન્ટરી મેનેજર: સ્ટોક લેવલ અપડેટ કરે છે

મેમરી ટ્રીક

“PPMI” - Process order, Process payment, Manage inventory

પ્રશ્ન 3(અ) OR [3 ગુણ]

અનાલિસિસ અને ડિઝાઇન વર્ચ્યેનો તફાવત લખો

જવાબ

પાસું	અનાલિસિસ	ડિઝાઇન
ધ્યાન	સિસ્ટમે શું કરવું જોઈએ	સિસ્ટમ કેવી રીતે કામ કરશે
તબક્કો	જરૂરિયાત તબક્કો	ડિઝાઇન તબક્કો
આઉટપુટ	સમસ્યાની સમજ	સોલ્યુશન સ્ટ્રક્ચર
મોડેલ	યુઝ કેસ, જરૂરિયાતો	આઇટેક્ચર, કલાસ
દૃષ્ટિકોણ	વપરાશકર્તાનો દૃષ્ટિકોણ	ડેવલપરનો દૃષ્ટિકોણ
સ્તર	અમૂર્ત, સંકલ્પનાત્મક	નક્કર, વિગતવાર

- એનાલિસિસ: સમસ્યા-કેન્દ્રિત, જરૂરિયાતોની સમજ
- ડિઝાઇન: સોલ્યુશન-કેન્દ્રિત, સિસ્ટમ આર્કિટેક્ચર બનાવવું

મેમરી ટ્રીક

"એનાલિસિસ=શું, ડિઝાઇન=કેવી રીતે"

પ્રશ્ન 3(બ) OR [4 ગુણ]

બેઝિક COCOMO મોડલ દ્વારા પ્રોજેક્ટ એસ્ટીમેશન પદ્ધતિ સમજાવો

જવાબ

બેઝિક COCOMO મોડલ: કોડની લાઈન આધારે સોફ્ટવેર ડેવલપમેન્ટ પ્રયત્નનો અંદાજ કાઢે છે.

સૂત્ર:

- Effort = $a \times (KLOC)^b person - months$
- Time = $c \times (Effort)^d months$

પ્રોજેક્ટ પ્રકારો:

પ્રકાર	a	b	c	d	વર્ણન
ઓર્ગનિક	2.4	1.05	2.5	0.38	નાની, અનુભવી ટીમ
સેમી-ડિટેચ્ડ	3.0	1.12	2.5	0.35	મધ્યમ કદ, મિશ્ર ટીમ
એમ્બેડ્ડ	3.6	1.20	2.5	0.32	જટિલ, કડક મર્યાદાઓ

પગલાં:

- KLOC (કોડની હજારો લાઈન) નો અંદાજ કાઢવો
- પ્રોજેક્ટ પ્રકાર ઓળખવો (organic/semi-detached/embedded)
- ચોંચ ગુણાંકો લાગુ કરવા
- પ્રયત્ન અને ડેવલપમેન્ટ સમયની ગણતરી કરવી

ઉદાહરણ: 10 KLOC ઓર્ગનિક પ્રોજેક્ટ

- Effort = $2.4 \times (10)^{1.05} = 25.2 person - months$
- Time = $2.5 \times (25.2)^{0.38} = 8.4 months$

મેમરી ટ્રીક

"OSE" - Organic, Semi-detached, Embedded

પ્રશ્ન 3(ક) OR [7 ગુણ]

લાઈબ્રેરી મેનેજમેન્ટ સિસ્ટમ માટે કલાસ ડાયગ્રામ દોરો અને સમજાવો

જવાબ

લાઈબ્રેરી મેનેજમેન્ટ સિસ્ટમ માટે કલાસ ડાયગ્રામ:

```
classDiagram
class Library {
    +name: String
    +address: String
    +addBook()
    +removeBook()
    +searchBook()
}

class Book {
    +bookId: String
    +title: String
    +author: String
}
```

```

+ISBN: String
+isAvailable: Boolean
+getDetails()
\}

class Member \{
    +memberId: String
    +name: String
    +email: String
    +phone: String
    +issueBook()
    +returnBook()
\}

class Transaction \{
    +transactionId: String
    +issueDate: Date
    +returnDate: Date
    +fine: Double
    +calculateFine()
\}

Library "1" o{-{-} "many" Book}
Member "1" o{-{-} "many" Transaction}
Book "1" o{-{-} "many" Transaction}

```

સંબંધો:

સંબંધ	વર્ણન	મલિટાલિસ્ટી
લાઇબ્રેરી-બુક	લાઇબ્રેરીમાં પુસ્તકો છે	1 થી ઘણા
મેમ્બર-ટ્રાન્జેક્શન	મેમ્બરના ટ્રાન્જેક્શન છે	1 થી ઘણા
બુક-ટ્રાન્જેક્શન	પુસ્તક ટ્રાન્જેક્શનમાં સામેલ	1 થી ઘણા

મુખ્ય લક્ષણો:

- એટ્રિબ્યુટ્સ: દરેક કલાસના ડેટા સર્બ્યો
- મેથડ્સ: કલાસ ડેટા પર કામ કરતા ફુંક્શન
- એસોસિએશન: કલારો વર્ચેના સંબંધો બતાવે છે કે તેઓ કેવી રીતે કિયાપ્રતિક્રિયા કરે છે

મેમ્બરી ટ્રીક

“LBMIT” - Library, Book, Member, Transaction

પ્રશ્ન 4(અ) [3 ગુણ]

પ્રોજેક્ટ સાઈઝ નક્કી કરવાના મેટ્રિક્સની યાદી બનાવી તેની વ્યાખ્યા લખો

જવાબ

પ્રોજેક્ટ સાઈઝ એર્ટીમેશન મેટ્રિક્સ:

મેટ્રિક	વ્યાખ્યા	ઉપયોગ
લાઇન્સ ઓફ કોડ (LOC)	એક્ઝિક્યુટેબલ કોડ લાઇનની ગણતરી	પરંપરાગત સાઈઝિંગ
ફંક્શન પોઇન્ટ્સ (FP)	કાર્યક્ષમતા આધારિત માપ	ભાષા-સ્વતંત્ર
ફીચર પોઇન્ટ્સ	વિસ્તૃત ફંક્શન પોઇન્ટ્સ	રિયલ-ટાઇમ સિસ્ટમ
ઓફજેક્ટ પોઇન્ટ્સ	ઓફજેક્ટ અને મેથડ્સની ગણતરી	ઓફજેક્ટ-ઓરિએટેડ સિસ્ટમ
યુઝ કેસ પોઇન્ટ્સ	યુઝ કેસ જટિલતા આધારિત	જરૂરિયાત-આધારિત

ફુંક્શન પોઇન્ટ્સ ઘટકો:

- એક્સ્ટાર્નલ ઇનપુટ્સ: ડેટા એન્ટ્રી સ્ક્રીન
 - એક્સ્ટાર્નલ આઉટપુટ્સ: રિપોટ્સ, મેસેજ
 - એક્સ્ટાર્નલ ઇન્કવાયરીઝ: ઇન્ટેક્ટિવ કવેરીઝ
 - ઇન્ટરનલ ફૂઈલ્સ: માર્ક્ટર ફૂઈલ્સ
 - એક્સ્ટાર્નલ ઇન્ટરફેસ: શેર કરેલ ડેટા

કાયદાઓ: વહેલું અનુમાન, ટેકનોલોજી-સ્વતંત્ર, માનકીકૃત અભિગમ

ਮੇਮਰੀ ਟ੍ਰੀਕ

“LFFOU” - LOC, Function Points, Feature Points, Object Points, Use Case Points

પ્રશ્ન 4(બ) [4 ગુણ]

જોખમની ઓળખને વિસ્તારથી સમજાવો

ଜ୍ଵାବ

જોખમ ઓળખ: પ્રોજેક્ટની સફળતાને અસર કરી શકે તેવા સંભવિત જોખમોને શોધવા, ઓળખવા અને વર્ણવવાની પ્રક્રિયા.
જોખમ કટેગરીઓ:

કેટેગરી	ઉદાહરણો	પ્રભાવ
ટેકનિકલ	નવી ટેકનોલોજી, જટિલતા	ડેવલપમેન્ટ વિલંબ
પ્રોજેક્ટ	શેડ્યુલ, બજેટ મર્યાદાઓ	કોસ્ટ ઓવરરન
વિઝનેસ	માર્કેટ ફેરફારો, સ્પર્ધા	પ્રોજેક્ટ રદીકરણ
બાહ્ય	વેન્ડર મુદ્દાઓ, નિયમો	નિભરતાઓ

ଓଇନ୍‌ଫ୍ରେମ୍‌ଵିଦ୍ୟା

- બેનસ્ટોર્મિંગ: જોખમો ઓળખવા માટે ટીમ ચર્ચા
 - ચેકલિસ્ટ: માનક જોખમ કેટેગરીઝની સમીક્ષા
 - એક્સપર્ટ જજામેન્ટ: અનુભવ આધારિત ઓળખ
 - SWOT અનાલિસિસ: શક્તિઓ, નવણાઈઓ, તકો, ઘમકીયો

રિસ્ક રજિસ્ટર: ઓળખાયેલ જોખમો સાથેનો દસ્તાવેજ જેમાં છે:

- જોખમ વર્ણન
 - ઘટનાની સંભાવના
 - પ્રભાવની ગંભીરતા
 - જોખમ કેટેગરી
 - જવાબદાર વ્યક્તિ

ਮੈਮਰੀ ਵੀਕ

“TPBE” - Technical, Project, Business, External risks

પ્રશ્ન 4(ક) [૭ ગુણ]

તમારી પસંદની કોઇ સિસ્ટમ માટે Gantt Chart દોડો

ଜୟାମ

ઓનલાઈન બેંકિંગ સિસ્ટમ માટે જેન્ર ચાર્ટ:

કાર્ય અઠવાડિયું 1 અઠવાડિયું 2 અઠવાડિયું 3 અઠવાડિયું 4 અઠવાડિયું 5 અઠવાડિયું 6 અઠવાડિયું 7 અઠવાડિયું 8

જુદ્ગિયાત પરિવહન પરિવહન

७३८

શ્રીષ્ટા

સિસ્ટમ	સિસ્ટમ ડિઝાઇન	સિસ્ટમ ડિઝાઇન
ડિ-ટેટાબેઝ	ડિ-ટેટાબેઝ	ડિ-ટેટાબેઝ
UI	UI ડેવલપમેન્ટ	UI ડેવલપમેન્ટ
ડેવલ-પમેન્ટ	બેકઅન્ડ	ડેવલ-પમેન્ટ
ટેસ્ટિંગ	ડેવલ-પમેન્ટ	ટેસ્ટિંગ
ડિપ્લોયમેન્ટ		ડિપ્લોયમેન્ટ

પ્રોજેક્ટ કાર્યો:

કાર્ય	અવધિ	નિર્ભરતાઓ	સંસાધનો
જરૂરિયાત વિશ્લેષણ	2 અઠવાડિયા	કોઈ નહીં	બિજનેસ એનાલિસ્ટ
સિસ્ટમ ડિઝાઇન	2 અઠવાડિયા	જરૂરિયાતો	સિસ્ટમ ડિઝાઇનર
ડિટાબેઝ ડિઝાઇન	2 અઠવાડિયા	સિસ્ટમ ડિઝાઇન	ડિટાબેઝ ડિઝાઇનર
UI ડેવલપમેન્ટ	2 અઠવાડિયા	સિસ્ટમ ડિઝાઇન	UI ડેવલપર
બેકઅન્ડ ડેવલપમેન્ટ	2 અઠવાડિયા	ડિટાબેઝ ડિઝાઇન	બેકઅન્ડ ડેવલપર
ટેસ્ટિંગ	2 અઠવાડિયા	UI + બેકઅન્ડ	QA ટેસ્ટર
ડિપ્લોયમેન્ટ	2 અઠવાડિયા	ટેસ્ટિંગ	DevOps એન્જિનિયર

ફાયદાઓ: દ્રશ્ય પ્રગતિ ટ્રેકિંગ, સંસાધન ફાળવણી, નિર્ભરતા વ્યવસ્થાપન

મેમ્સ્રી ટ્રીક

"RSDUBtd" - Requirements, System design, Database, UI, Backend, Testing, Deployment

પ્રશ્ન 4(અ) OR [3 ગુણ]

પ્રોજેક્ટ મેનેજરની જવાબદારીઓની ચાહી બનાવો

જવાબ

પ્રોજેક્ટ મેનેજરની જવાબદારીઓ:

ક્ષેત્ર	જવાબદારીઓ
આપોઝન	પ્રોજેક્ટ પ્લાન બનાવવા, સ્કોપ વ્યાખ્યાપિત કરવો
સંગઠન	સંસાધનો ફાળવવા, ટીમ બનાવવી
નેતૃત્વ	ટીમને પ્રેરણા આપવી, સંઘર્ષ ઉકેલવો
નિયંત્રણ	પ્રગતિ મોનિટર કરવી, ફેરફારો વ્યવસ્થિત કરવા
કોમ્પ્યુનિકેશન	હિતધારક અપડેટ્સ, ટીમ કોર્ડિનેશન
રિસ્ક મેનેજમેન્ટ	જોખમો ઓળખવા અને શમન કરવા

મુખ્ય પ્રવૃત્તિઓ:

- પ્રોજેક્ટ શરૂઆત: ઉદ્દેશ્યો અને મર્યાદાઓ વ્યાખ્યાપિત કરવા
- શેડ્યુલ મેનેજમેન્ટ: ટાઈમલાઇન બનાવવી અને જાળવવી
- બજેટ નિયંત્રણ: ખર્ચ અને વ્યય મોનિટર કરવા
- ગુણવત્તા આશ્વાસન: ડિલિવરેબલ સ્ટાન્ડર્ડ સુનિશ્ચિત કરવા
- ટીમ મેનેજમેન્ટ: ટીમ સભ્યોનું નેતૃત્વ અને વિકાસ

પ્રશ્ન 4(બ) OR [4 ગુણ]

જોખમ આકારણીને વિસ્તારથી સમજાવો

જવાબ

જોખમ આકારણી: પ્રોજેક્ટની સફળતા પર તેમની સંભાવના અને પ્રભાવ નક્કી કરવા માટે ઓળખાયેલ જોખમોનું મૂલ્યાંકન કરવાની પ્રક્રિયા.

આકારણી ઘટકો:

ઘટક	સ્કેલ	વર્ણન
સંભાવના	1-5 અથવા %	જોખમ ઘટનાની સંભાવના
પ્રભાવ	1-5 અથવા \$	જો જોખમ થાય તો તીવ્રતા
રિસ્ક સ્કોર	$P \times I$	એકંદર જોખમ પ્રાથમિકતા

જોખમ આકારણી મેટ્રિક્સ:

સંભાવના/પ્રભાવ	નીચું (1)	મધ્યમ (2)	ઉચ્ચ (3)
નીચું (1)	1	2	3
મધ્યમ (2)	2	4	6
ઉચ્ચ (3)	3	6	9

આકારણી તકનીકો:

- ગુણાત્મક આકારણી: વર્ણનાત્મક સ્કેલ (ઉચ્ચ/મધ્યમ/નીચું)
- માત્રાત્મક આકારણી: સંપ્રાત્મક મૂલ્યો અને ગણતરીઓ
- એક્સપર્ટ જજ્ઞમેન્ટ: અનુભવ આધારિત મૂલ્યાંકન
- ઐતિહાસિક ડેટા: ભૂતકાળના પ્રોજેક્ટ વિશ્લેષણ

જોખમ વર્ગીકરણ:

- ઉચ્ચ જોખમ (7-9): તાત્કાલિક ધ્યાન જરૂરી
- મધ્યમ જોખમ (4-6): મોનિટર કરવું અને શમન આયોજન કરવું
- નીચું જોખમ (1-3): સ્વીકારવું અથવા ન્યૂનતમ શમન

પ્રશ્ન 4(ક) OR [7 ગુણ]

તમારી પસંદની કોઇ સિસ્ટમ માટે સ્પ્રિન્ટ બર્ન ડાઉન ચાર્ટ દોરો

જવાબ

E-commerce મોબાઇલ એપ માટે સ્પ્રિન્ટ બર્ન ડાઉન ચાર્ટ (2-અઠવાડિયાનો સ્પ્રિન્ટ):

```

| 
40 +{-{-}{-}*}
|   {}
35 +   *
|   {}
30 +   *
|   {}
25 +     *{-{-}{-}*}
|   {}
20 +       *

```

```

|          {}
15 +
|          *
|          {}
10 +
|          *
|          {}
5 +
|          *
|          {}
0 +\_\_\
1 2 3 4 5 6 7 8 9 10
* = {-{-}}{-} = }

```

स्प्रॅन्ट विगतीः

દિવસ	આદર્શ બાકી	વાસ્તવિક બાકી	પૂર્ણ થયેલ કાર્ય
દિવસ 1	36	40	સ્પ્રિન્ટ પ્લાનિંગ
દિવસ 2	32	35	યુઝર લોગિન ફીચર
દિવસ 3	28	30	પ્રોડક્ટ કેટાલોગ
દિવસ 4	24	25	શોપિંગ કાર્ટ
દિવસ 5	20	25	API મુદ્દાથી અવરોધ
દિવસ 6	16	20	પેમેન્ટ એક્કિકરણ
દિવસ 7	12	15	ઓર્ડર મેનેજમેન્ટ
દિવસ 8	8	10	ટેસ્ટિંગ અને ફિક્સ
દિવસ 9	4	5	અંતિમ ટેસ્ટિંગ
દિવસ 10	0	0	સ્પ્રિન્ટ પૂર્ણ

મુખ્ય અંતર્દૃષ્ટિ:

- ઢાળ: આદર્શ સાથે સરખામણીએ પ્રગતિ દર
 - સપાટ વિસ્તારો: અવરોધિત કાર્ય અથવા સ્કોપ ફેરફારો
 - આદર્શથી નીચે: શેડ્જૂલ આગળ
 - આદર્શથી ઉપર: શેડ્જૂલ પાછળ

ਮੇਮਰੀ ਟ੍ਰੀਕ

``DABC" - Days, Actual, Burn-down, Chart

પ્રશ્ન 5(અ) [૩ ગુણ]

કોડ રિવ્યુ તકનીકની યાદી બનાવી કોઈ એક સમજાવો

ଜୟାମ

કોડ રિવ્યુ તકનીકો:

તકનીક	વર્ણન	સહભાગીઓ
કોડ વેક્ચ્યુ	લેખક દ્વારા અનૌપચારિક સમીક્ષા	લેખક + સમીક્ષકો
કોડ ઇન્સ્પેક્શન	ઔપચારિક, વ્યવસ્થિત સમીક્ષા	પ્રશિક્ષિત નિરીક્ષકો
પીએર રિવ્યુ	સાથીદાર કોડ તપાસે છે	ડેવલપર સાથીદારો
ટુલ-આધારિત રિવ્યુ	સ્વચાલિત વિશ્લેષણ	ટુલ્સ + ડેવલપર્સ

કોડ ઇન્સ્પેક્શન સમજાવેલ:

પ્રક્રિયા:

- આચ્યોજન: કોડ પસંદ કરવો, ભૂમિકાઓ સૌંપવી
- જાંખી: લેખક કોડ સ્ટ્રક્ચર સમજાવે છે
- તૈયારી: કોડની વ્યક્તિગત સમીક્ષા
- ઇન્સ્પેક્શન મીટિંગ: ગ્રૂપ કોડ તપાસે છે
- રિવીઝ: ઓળખાયેલ ખામીઓ ઠીક કરવી
- ફીલો-અપ: સુધારાઓ ચકાસવા

ભૂમિકાઓ:

- મોડેટર: ઇન્સ્પેક્શન પ્રક્રિયાનું નેતૃત્વ
- લેખક: કોડ ડેવલપર, લોજિક સમજાવે છે
- સમીક્ષકો: ખામીઓ અને મુદ્દાઓ શોધે છે
- રેકૉર્ડર: તારણો દસ્તાવેજુકૃત કરે છે

ફાયદાઓ: ઉચ્ચ ખામી શોધ દર, જ્ઞાન શેરિંગ, સુધારેલ કોડ ગુણવત્તા

મેમરી ટ્રીક

“CWIP” - Code Walkthrough, Inspection, Peer review

પ્રશ્ન 5(બ) [4 ગુણ]

ઓનલાઈન શોપિંગ સિસ્ટમ માટે ટેસ્ટ કેસ તૈયાર કરો

જવાબ

ઓનલાઈન શોપિંગ સિસ્ટમ માટે ટેસ્ટ કેસ:

ટેસ્ટ કેસ ID	ટેસ્ટ દૃશ્ય	ટેસ્ટ સ્ટેપ્સ	અપેક્ષિત પરિણામ
TC001	વપરાશકર્તા નોંધણી	1. માન્ય વિગતો દાખલ કરો2. રજિસ્ટર ક્લિક કરો	એકાઉન્ટ સફળતાપૂર્વક બનાવ્યું
TC002	વપરાશકર્તા લોગિન	1. વપરાશકર્તાનામ/પાસવર્ડ દાખલ કરો2. લોગિન ક્લિક કરો	વપરાશકર્તા લોગ ઇન થયો
TC003	કાર્ટમાં ઉમેરો	1. પ્રોડક્ટ પસંદ કરો2. કાર્ટમાં ઉમેરો ક્લિક કરો	પ્રોડક્ટ કાર્ટમાં ઉમેર્યું
TC004	ચેકઆઉટ પ્રક્રિયા	1. કાર્ટમાં જાઓ2. ચેકઆઉટ ક્લિક કરો3. પેમેન્ટ વિગતો દાખલ કરો	ઓર્ડર સફળતાપૂર્વક મૂક્યો

વિગતવાર ટેસ્ટ કેસ ઉદાહરણ:

ટેસ્ટ કેસ ID: TC003 ટેસ્ટ ટાઇટલ: શોપિંગ કાર્ટમાં પ્રોડક્ટ ઉમેરવું પ્રી-કન્ડિશન: વપરાશકર્તા લોગ ઇન છે, પ્રોડક્ટ ઉપલબ્ધ છે ટેસ્ટ સ્ટેપ્સ:

- પ્રોડક્ટ કેટાલોગ પર નેવિગેટ કરો
- પ્રોડક્ટ પસંદ કરો
- જગ્યો પસંદ કરો
- “કાર્ટમાં ઉમેરો” બટન ક્લિક કરો

અપેક્ષિત પરિણામ: સાચા જથ્થા અને કિમત સાથે પ્રોડક્ટ કાર્ટમાં દેખાય છે પોસ્ટ-કન્ડિશન: કાર્ટ કાઉન્ટ વધે છે, કુલ રકમ અપડેટ થાય છે

મેમરી ટ્રીક

“RAULC” - Registration, Authentication, User cart, Login, Checkout

પ્રશ્ન 5(ક) [7 ગુણ]

વ્હાઇટ બોક્સ ટેકનિકની વ્યાખ્યા કરો. વિવિધ વ્હાઇટ બોક્સ તકનીકની સૂચિ બનાવો. કોઈપણ બે સમજાવો

જવાબ

વહાઇટ બોક્સ ટેસ્ટિંગ વ્યાખ્યા: આંતરિક કોડ સ્ટ્રક્ચર, લોજિક પાથ અને અમલીકરણ વિગતોની તપાસ કરતી ટેસ્ટિંગ તકનીક.
વહાઇટ બોક્સ તકનીકો:

તકનીક	કવરેજ કાઇટેરિયા	હેતુ
સ્ટેટમેન્ટ કવરેજ	બધા સ્ટેટમેન્ટ એક્ઝિક્યુટ બધી બાન્ચ લેવાય	બેસિક કોડ કવરેજ
બાન્ચ કવરેજ	બધી પાથ એક્ઝિક્યુટ	નિર્ણય ટેસ્ટિંગ
પાથ કવરેજ	બધા શરતો ટેસ્ટ	સંપૂર્ણ ફ્લો ટેસ્ટિંગ
કન્ડિશન કવરેજ	બધી શરતો ટેસ્ટ	લોજિકલ એક્સપ્રેશન ટેસ્ટિંગ
લૂપ ટેસ્ટિંગ	બધા લૂપ વેરિએશન	પુનરાવર્તક સ્ટ્રક્ચર ટેસ્ટિંગ

1. સ્ટેટમેન્ટ કવરેજ: કોડમાં દરેક એક્ઝિક્યુટેબલ સ્ટેટમેન્ટ ઓછામાં ઓછું એક વાર એક્ઝિક્યુટ થાય તેની ખાતરી કરે છે.
સૂત્ર: (એક્ઝિક્યુટ થયેલ સ્ટેટમેન્ટ / કુલ સ્ટેટમેન્ટ) × 100%

ઉદાહરણ:

```
if (x > 0)      //    1
    y = x + 1;  //    2
else
    y = x - 1;  //    3
z = y * 2;       //    4
```

ટેસ્ટ કેસ: x = 5 (સ્ટેટમેન્ટ 1,2,4 કવર કરે), x = -1 (સ્ટેટમેન્ટ 1,3,4 કવર કરે) કવરેજ: 100% સ્ટેટમેન્ટ કવરેજ હાંસલ

2. બાન્ચ કવરેજ: નિર્ણય બિદ્ધાયોની દરેક બાન્ચ (true/false) એક્ઝિક્યુટ થાય તેની ખાતરી કરે છે.

ઉદાહરણ:

```
if (a > b && c > d)    //
    result = 1;        // True
else
    result = 0;        // False
```

ટેસ્ટ કેસ:

- a=5,
b=3,
c=7,
d=2 (true બાન્ચ)
- a=1,
b=3,
c=7,
d=2 (false બાન્ચ)

ફાયદાઓ: સ્ટેટમેન્ટ કવરેજ કરતાં ઉચ્ચ ખામી શોધ

મેમરી ટ્રીક

“SBPCL” - Statement, Branch, Path, Condition, Loop

પ્રશ્ન 5(અ) OR [3 ગુણ]

સોફ્ટવેર ડોક્યુમેન્ટેશન સમજાવો

જવાબ

સોફ્ટવેર ડોક્યુમેન્ટેશન: સોફ્ટવેર સિસ્ટમ, તેની ડિઝાઇન, અમલીકરણ અને ઉપયોગનું વર્ણન કરતી લેખિત સામગ્રી.
ડોક્યુમેન્ટેશનના પ્રકારો:

પ્રકાર	હેતુ	પ્રેક્ષકો
આંતરિક ડોક્યુમેન્ટેશન	કોડ સમજ	ડેવલપર્સ
બાહ્ય ડોક્યુમેન્ટેશન	સિસ્ટમ ઉપયોગ	વપરાશકર્તાઓ, ઓપરેટર્સ
સિસ્ટમ ડોક્યુમેન્ટેશન	ડિઝાઇન અને આર્કિટેક્ચર	જાળવણીકર્તાઓ
વપરાશકર્તા ડોક્યુમેન્ટેશન	ઓપરેશન સૂચનાઓ	અંતિમ વપરાશકર્તાઓ

આંતરિક ડોક્યુમેન્ટેશન:

- ટિપ્પણીઓ: કોડ લોજિક અને હેતુ સમજાવે છે
- કોડ સ્ટ્રક્ચર: કલાસ અને મેથ્ડ વર્ણનો
- ડિઝાઇન તક: શા માટે ચોક્કસ અભિગમ પરસંદ કર્યો

બાહ્ય ડોક્યુમેન્ટેશન:

- વપરાશકર્તા મેન્યુઅલ: સ્ટેપ-બાય-સ્ટેપ ઉપયોગ સૂચનાઓ
- ઇન્સ્ટોલેશન ગાઇડ: સેટઅપ પ્રક્રિયાઓ
- API ડોક્યુમેન્ટેશન: ઇન્ટરફેસ સ્પેસિફિકેશન

ફાયદાઓ: સરળ જાળવણી, જાન સ્થાનાંતરણ, ઘટાડેલ તાલીમ સમય

મેમરી ટ્રીક

"IESU" - Internal, External, System, User documentation

પ્રશ્ન 5(બ) OR [4 ગુણ]

ATM સિસ્ટમ માટે 4 ટેસ્ટ કેસ બનાવો

જવાબ

ATM સિસ્ટમ માટે ટેસ્ટ કેસ:

ટેસ્ટ કેસ ID	ટેસ્ટ દૃશ્ય	ટેસ્ટ સ્ટેપ્સ	અપેક્ષિત પરિણામ
TC001	માન્ય PIN એન્ટ્રી	1. કાર્ડ દાખલ કરો2. સાચો PIN દાખલ કરો3. Enter દબાવો	મુખ્ય મેનુમાં પ્રવેશ મળ્યો
TC002	અમાન્ય PIN એન્ટ્રી	1. કાર્ડ દાખલ કરો2. ખોટો PIN દાખલ કરો3. Enter દબાવો	"અમાન્ય PIN" સંદેશ દેખાય છે
TC003	રોકડ ઉપાડ	1. સફળતાપૂર્વક લોજિન કરો2. "રોકડ ઉપાડ" પસંદ કરો3. રકમ દાખલ કરો4. પુષ્ટ કરો	રોકડ આપવામાં આવી, બેલેન્સ અપડેટ થયું
TC004	અપૂરતું બેલેન્સ	1. સફળતાપૂર્વક લોજિન કરો2. "રોકડ ઉપાડ" પસંદ કરો3. બેલેન્સ કરતાં વધુ રકમ દાખલ કરો	"અપૂરતું બેલેન્સ" સંદેશ

વિગતવાર ટેસ્ટ કેસ:

ટેસ્ટ કેસ ID: TC003 ટેસ્ટ વર્ણન: પૂરતા બેલેન્સ સાથે રોકડ ઉપાડવી પ્રી-કન્ડિશન: માન્ય ATM કાર્ડ, પૂરતું એકાઉન્ટ બેલેન્સ ટેસ્ટ ડેટા: PIN=1234, ઉપાડની રકમ=₹1000, એકાઉન્ટ બેલેન્સ=₹5000

પોસ્ટ-કન્ડિશન: એકાઉન્ટ બેલેન્સ ₹1000 ઘટાડ્યું, ટ્રાન્ઝેક્શન રેકૉર્ડ થયું

મેમરી ટ્રીક

"VPCI" - Valid PIN, PIN error, Cash withdrawal, Insufficient funds

પ્રશ્ન 5(ક) OR [7 ગુણ]

બ્લેક બોક્સ ટેસ્ટિંગ પદ્ધતિની સૂચિ બનાવો. તેને ફંક્શનલ ટેસ્ટિંગ કેમ કહેવાય છે તે સમજાવો. થા કોઈ પણ બે પદ્ધતિ આકૃતિ સાથે વણવો

જવાબ

બ્લેક બોક્સ ટેસ્ટિંગ પદ્ધતિઓ:

પદ્ધતિ	હેતુ	ઇનપુટ ફોક્સ
સમકક્ષ વિભાજન બાઉન્ડરી વેલ્યુ એનાલિસિસ	ઇનપુટને વર્ગોમાં વહેંચ્યતું સીમા મૂલ્યોની ટેસ્ટ	માન્ય/અમાન્ય વિભાજન સીમા શરતો

ડિસ્ટિગ્ન ટેબલ ટેસ્ટિંગ	જટિલ બિજનેસ નિયમો	બહુવિધ ઇનપુટ સંચોળનો
સ્ટેટ ટ્રાન્ઝિશન ટેસ્ટિંગ	સ્ટેટ આધારિત સિસ્ટમ	સ્ટેટ ફેરફારો
ચુઝ કેસ ટેસ્ટિંગ	કાર્યોત્તમક દૃષ્ટ્યો	વપરાશકર્તા કિયાપ્રતિક્રિયા
ઓરર ગેસિંગ	અનુભવ આધારિત ટેસ્ટિંગ	સંભવિત ભૂલ શરતો

શા માટે ફુંક્શનલ ટેસ્ટિંગ કહેવાય છે? બ્લેક બોક્સ ટેસ્ટિંગ સિસ્ટમ શું કરે છે પર ધ્યાન આપે છે તે કેવી રીતે કામ કરે છે તેનાથી વિપરીત. તે આંતરિક કોડ સ્ટ્રક્ચરનું જ્ઞાન વિના ઇનપુટ અને અપેક્ષિત આઉટપુટ ટેસ્ટ કરીને કાર્યોત્તમક આવશ્યકતાઓને માન્ય કરે છે.

1. સમકક્ષ વિભાજન:

: (0{-120})

:	:
18{-65	0
v	v
[]	[]

ઉદાહરણ: જોબ એપ્લિકેશન માટે વય વેલિડેશન

- માન્ય વિભાજન: 18-65 વર્ષ
- અમાન્ય વિભાજન: <0, 0-17, 66-120, >120
- ટેસ્ટ કેસ: દરેક વિભાજનમાંથી એક (દા.ત., 25, -5, 10, 70, 130)

2. બાઉન્ડરી વેલ્યુ અનાલિસિસ:

: (0{-100})

{-1	0
v	v
[]	[]

ઉદાહરણ: વિદ્યાર્થી સ્કોર વેલિડેશન (0-100)

- ટેસ્ટ મૂલ્યો: -1, 0, 1, 50, 99, 100, 101
- ફોકસ: સીમાની અંદર અને બહાર
- તક્ક: મોટાભાગની ભૂલો રીમા પર થાય છે

ફાયદાઓ:

- સ્વતંત્રતા: પ્રોગ્રામિંગ જાનની આવશ્યકતા નથી
- વપરાશકર્તા દૃષ્ટિકોણ: વપરાશકર્તાના દૃષ્ટિકોણથી ટેસ્ટ
- જરૂરિયાત વેલિડેશન: કાર્યોત્તમક સ્પેસિફિકેશન ચકાસે છે

મેમરી ટ્રીક

“EBDSUE” - Equivalence, Boundary, Decision, State, Use case, Error guessing